Law breaking trading algorithms: Emergence and deterrence

Henry Ashton
[redacted]

A dissertation submitted in partial fulfilment
of the requirements for the degree of
Doctor of Philosophy
of
University College London.

Computer Science
University College London

September 12, 2022

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

Abstract

This thesis demonstrates that trading algorithms trained through Reinforcement Learning will learn to manipulate prices (thereby breaking the law) through a process called *spoofing* in a fully functioning limit order book environment. The regulatory definition of Spoofing requires the establishment of intent on part of the accused: it is defined by the US CFTC as the placement of orders with the intent to cancel them. This needs to be defined for auto-didactic algorithms where behaviour emerges somewhat independently from the programmer. I propose a high-level definition informed by current law then test to see whether it matches with a laypeople's natural understanding of the concept. Finally, I implement a constrained learning method in Reinforcement Learning using an appropriate definition of intent to cancel which allow auto-didactic trading algorithms to be trained and deployed safely without the risk of spoofing behaviour emerging.

This subject is important because algorithmic trading leads other areas in the degree of agency that algorithmic actors are permitted. The simple pursuit of high-level objectives like profit maximisation can result in behaviour that contradicts the law. Without a method of encoding laws within the training and testing process, algorithms will likely learn to break laws when it is rational to do so.

The research comprises:

1. **Platform: BUCLSE** A Limit Order Book simulation environment pro-

grammed in Python. It is flexible enough to admit multiple types of traders of varying ability ranging from simple zero intelligence traders to complex reinforcement learning based traders. Having a limit order book simulator that accurately reflects the market impact of a trader's actions is a prerequisite for investigating market manipulation risk. As such it is an example of a digital twin where emergent behaviour can be studied and controlled safely.

2. **Emergence of spoofing in a RL LOB** In this experiment we demonstrate that a Reinforcement learning agent, fed with a simple state space representation of a limit order book, can learn to manipulate the market to its favour. We test a variety of RL techniques including Deep Q learning and Dyna Q using both a Conditional Variational Autoencoder and a traditional tabular based model. The resulting strategies are distilled into a simple decision tree for interpretation.
3. **A definition of intent for algorithms** In this chapter we construct various definitions of intent suitable for algorithms from the body of research that concerns intent in common law.
4. **Testing a definition of intent for algorithms on laypeople** Here we test laypeople's attitude towards intent in an algorithm and test a simple definition of intent. We contrast judgements of intent of AI with a Human and judgements made with and without a provided definition of intent over various experimental configurations.
5. **A method to identify and restrict intent to cancel in a simple queuing example** For autonomous algorithms to be deployed in regulated environments they need to be law abiding. Building on the definitions introduced earlier we discuss the minimum requirements for a trading algorithm to have intent, and illustrate how to measure and control intent using a shield in a minimal queuing game example.

The thesis makes the following contributions to science:

- Establishes that the risk of market manipulation is 'foreseeable' for anyone wishing to deploy RL-trained trading algorithm in a Limit Order Book. This could be important in deciding how legal liability is imposed in court.
- Defines the concept of *intent* for autonomous algorithms which is based on existing legal principles. This allows the testing of trading algorithms for compliance purposes and is necessary for their safe training and deployment in the market.
- Tests how the concept of intent in autonomous algorithms is received by laypeople. This is important because juries are comprised of laypeople in the United Kingdom amongst other common law countries
- Presents a practical method to define 'intent to cancel' along with a method in Reinforcement Learning to create trading policies that do not exhibit intent to cancel at the point of order placement and therefore do not spoof.

Acknowledgements

Computer Science: Firstly hats off to Philip Treleaven for guiding me diligently through the process of writing a PhD. Without his constant encouraging emails and calls I would not have completed this. Paolo Barucca for agreeing to be my second supervisor and our work together on Algorithmic trading. Jun Wang for encouraging my interest in reinforcement learning. Yonita Carter and Gina Baidoo for keeping everything running smoothly.

Law: Many thanks for the encouragement given to me by Mark DSouza and Kevin Toh at UCL Law. Alex Sarch at Surrey for putting me right on recklessness in MPC. John Child at Birmingham for inchoate intent.

Psychology Heartfelt thanks to David Lagnado and Matija Franklin at UCL Causal cognition for all their enthusiasm and the research we have done together.

Other Schools Thanks to Vladimir Deineko at Warwick for getting me into academic research, Godfrey Keller and Howard Smith at Oxford for their support over the years and Alan Graham at Portsmouth for seeding the idea of Machine Learning into my head as a topic for research.

Personal Cynthia and Richard - Thank you for all your love and the sacrifices you've made to support me. Eddie thank you for your company and for letting me know when to finish. Finally, all my love and gratitude to Louise - without you I would be nowhere.

Impact Statement

As the aptitude of algorithmic actors increases so will their autonomy and agency. Important legal questions need to be asked about what constitutes a crime when algorithms are perpetrators or co-conspirators and their behaviour is not generated by another human.

Algorithmic trading leads other areas in terms of algorithmic agency and autonomy yet the legal and regulatory framework of the sector requires urgent work in order to prevent autonomous algorithm 'crime' or law-breaking from becoming an issue. To date, regulation has typically been written to consider human actors; this can lead to ambiguities when the actors are algorithms. The stakes are high with American and European regulators imposing multi-million dollar fines on financial organisations for market abuse with increased regularity.

The experiments contained in this thesis cover the topics of prevention, interpretation and prosecution. They are relevant to a range of stakeholders across the spectrum of the finance industry from trader, to broker, exchange, regulator, law enforcement and prosecutor.

Whilst the setting of much of this thesis is that of algorithmic trading and the specific market manipulative practice of spoofing, the techniques developed in this thesis should be readily transferable into other areas governed by laws where algorithmic actors interact with us and each other, as and when they develop.

Author Biography

- Ashton, H. (2022b). Definitions of intent suitable for algorithms. *Artificial Intelligence and Law*
- Franklin, M., Ashton, H., Awad, E., and Lagnado, D. (2022a). Causal framework of AI responsibility. In *Fifth AAAI ACM conference on Artificial Intelligence, Ethics and Society*, Oxford, United Kingdom
- Ashton, H. and Franklin, M. (2022a). A method to check that participants are imagining artificial minds when ascribing mental states. In *Proceedings of 24th Human Computer Interaction International conference (HCII)*
- Franklin, M., Ashton, H., Gorman, R., and Armstrong, S. (2022b). Missing Mechanisms of Manipulation in the EU AI Act. *The International FLAIRS Conference Proceedings*, 35
- Ashton, H. (2022a). Defining and Identifying the Legal Culpability of Side Effects Using Causal Graphs. In *Workshop on AI Safety 2022 (SafeAI 2022) co-located with the Thirty Sixth AAAI Conference on Artificial Intelligence*
- Ashton, H. and Franklin, M. (2022b). The problem of behaviour and preference manipulation in AI systems. In *Workshop on AI Safety 2022 (SafeAI 2022) co-located with the Thirty Sixth AAAI Conference on Artificial Intelligence*
- Franklin, M., Ashton, H., Gorman, R., and Armstrong, S. (2022c). Recognising the importance of preference change: A call for a coordinated multidisciplinary research effort in the age of AI. *The AAAI-22 Workshop*

on AI For Behavior Change co-located with the Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI-22)

- Ashton, H. (2021c). What criminal and civil law tells us about Safe RL techniques to generate law-abiding behaviour. *Workshop on AI Safety 2021 co-located with the Thirty Fifth AAAI Conference on Artificial Intelligence*
- Ashton, H. (2021a). Causal Campbell-Goodhart's Law and Reinforcement Learning:. In *Proceedings of the 13th International Conference on Agents and Artificial Intelligence (ICAART)*, pages 67–73. SCITEPRESS - Science and Technology Publications
- Ashton, H. (2020). AI Legal Counsel to train and regulate legally constrained Autonomous systems. *IEEE International Conference on Big Data (Big Data)*, pages 2093–2098

Contents

List of Figures

List of Tables

List of Algorithms

Glossary

A2C An actor critic reinforcement learning algorithm developed from Mnih et al. (2016b). 226

Action Value (Q) Function In reinforcement learning, a function which expresses the value of choosing an action in a certain state and following some fixed policy thereafter. 244

Actor Critic In reinforcement learning, an algorithm which learns a representation of both a policy function and a value function. 258

actus reus Literally guilty act, the performative part of any crime. 105

inchoate offense Inchoate crimes are those which involve seeking or preparing to commit another crime. 129

MDP Markov Decision Process - A decision problem where an agent must choose a policy to maximise their total discounted reward in some system governed by a Markov process. 211

mens rea Literally guilty mind, the mental state required by an actor at the point of committing a crime. 105, 119

MPC Model Penal Code - An effort from the American law institute to codify and standardise legal terms for state criminal law in the US (The American Law Insitute, 2017). 202

Oblique intent The intentional status of foreseeable side-effects of actions taken also known as indirect intent. Introduced by Bentham (1823). 125

PPO Proximal Policy Optimisation - an Actor-Critic method of reinforcement learning introduced in Schulman et al. (2017) . 214

RL Reinforcement Learning. 197

UCB The name of an algorithm that balances exploration and exploitation in a sequential decision problem. 253

Ulterior intent A variety of intent concerning actions in the non-immediate future. 141

VAE Variational AutoEncoder - a generative deep learning technique which seeks to learn the distribution of some dataset and generate novel instances from that distribution. 249

Value Function In reinforcement learning, a function which expresses the value of being in any state and following some fixed policy thereafter. 244

Chapter 1

Introduction

This chapter contains an introduction to the topic of market manipulation, a brief look at the problem's background in published literature and the research questions that the thesis attempts to answer. Following this there is a summary of this thesis' scientific contributions before an outline of the rest of the work.

1.1 Motivation

Efficient price discovery is of fundamental importance to society. Exchange markets are a prime mechanism for this process to take place for many financial and physical assets. Unfortunately, actors have been found guilty of manipulating this process in the past for personal gain. Whilst society has responded by making certain trading practices illegal within markets [1], there remains a proportion of market participants who believe it profitable to act illegally. Since the laws exist, it is the duty of the regulator, the market operator and the market's participants to make sure they are not broken.

Algorithmic trading accounts for approximately 50% of volume within most modern markets (Hodge, 2019). Algorithms have superhuman capabilities of speed, memory and computational power in comparison with human traders.

[1] See FMSB (2018) for a comprehensive ontology.

They have been successful operating in modern markets for many years and there is no reason to believe this won't remain the case in the future. With these advantages, it comes as no surprise that they are being used to commit market abuse (Lin, 2016).

Over the past decade, developments in deep learning have allowed Deep Reinforcement Learning trained algorithms to surpass human mastery of certain games with minimal guidance from the experimenter. Development has been fast in recent years; a variety of Atari games were mastered by Mnih et al. (2015) using a single neural network with different game specific parameter weightings. Most recently Reed et al. (2022) have developed a generalist agent capable of many varied tasks using the same network weightings. Trading markets have obvious parallels with these games because they also have a limited state and action space that is already machine interpretable and a reward function in the form of profit. There is growing interest in applying these methods to develop a new generation of algorithmic traders similarly unguided by human hand in the hope that novel money-making trading strategies can be unearthed.

If human actors have found it rational to act illegally in regulated markets, it is a reasonable question to ask when self-taught algorithmic traders will follow suit. Algorithm led 'crime' or transgression poses questions about the responsibility and culpability of their original programmers and owners. How can market participants ensure that their auto-didactic algorithms are not market abusive by design? Do regulators and exchanges require a different set of tools to detect market manipulation instigated by algorithmic traders?

Spoofing is a simple example of a proscribed practice in most regulated markets. The Dodd-Frank Wall Street Reform and Consumer Protection Act, 7 U.S.C.A. §6c(a)(5)(C) defines it as "bidding or offering with the intent to cancel the bid or offer before execution". There are multiple motivations for a trader to place orders on an exchange without wanting to those orders to

be settled. Various terms for practice related to spoofing like pinging, quote-stuffing, layering have emerged in practice and literature but it is noticeable that regulators haven't seen fit to mention these in statute. This is understandable because they are already regulated under the definition of spoofing. Recognition of these terms is not appropriate because the motivation of the spoofer is irrelevant in labelling their behaviour prohibited.

This thesis will concentrate on spoofing as a deceptive practice whose motivation is to engender some predictable response from other market participants. This is achieved by the strategic posting of orders on a limit order book. The predictable response is triggered by making other market participants believe something about the state of the order book which is known to be false by the spoofer. Wang et al. (2018) draw parallels between this tactic and that of poisoning in adversarial learning Barreno et al. (2006), where agents inject false data into an opponent's data so as to induce a certain future response.

Chapter 4 will consider the specific case where spoofing behaviour emerges from an auto-didactic algorithm whose utility function is based on improving the exit price of an existing holding. This is spoofing motivated by simple trading profitability. For a putative spoofer wishing to profit from an increase (decrease) in best bid (ask), they will place a large order on the bid (ask) side. This should fool other market participants into believing the new large orders represent some informed information coming into the market. They will attempt to out bid (or under-ask) this large order. The spoofer hopes this reaction will improve best bid (ask), and they will take this new liquidity and cancel their original spoofing orders.

1.2 Background in Research

Definitions of market manipulation vary across markets globally but Lomnicka (2001) believe they are converging as markets become more interconnected.

The FMSB (2018) find there to be 13 'behavioural clusters' of market abuse. Amongst those, we have chosen to concentrate on an illegal behaviour known as Spoofing (or layering). Wang and Wellman (2017) define the practice as the submission of large orders to the market which the trader does not intend to be executed, but instead be interpreted by other market participants as indicators of imminent price movements. The trader can profit from the induced price movement by executing real orders on the opposite of the market before cancelling the original large order. Actually, regulators are not prescriptive about the motivation for spoofing simply defining it as the placement of orders with the intent to cancel them. In their guidance of anti-spoofing regulations, the CFTC (2013) suggests a non-exhaustive list of alternative motivations to spoof other than price movement. They might be to give a false impression of liquidity in a security, carrying out a denial-of-service attack on an exchange venue by overloading their systems or attempting to delay the execution of another participant's orders. Of the 13 behaviours detailed in FMSB (2018), we feel spoofing could manifest itself most easily in a simulated environment in conjunction with an auto-didactic trading algorithm because the requisite actions exist exclusively on exchange. This is classified as a *'trade-based'* manipulation by Allen and Gale (1992), Figure 1.1. A simple objective function - maximise trading profit within any trading period - is sufficient to rationalise spoofing behaviour whilst the required action space - place and cancel buy or sell orders of variable size and price on a market is relatively simple.

Continuing with Allen's taxonomy, *'Information based'* manipulation through the dissemination of false information could almost certainly be learnt by algorithm - see Lin (2016). The link between potentially bot generated tweets and stock performance has been explored by amongst others (Fan et al., 2020; Renault, 2017). Since realistic novel text can be generated by large language models such as GPT3 (Brown et al., 2020), the means to achieve this type of manipulation is within the scope of current algorithm capability. An integrated algorithm with the ability to write tweets and place orders would likely find

an effective information-based manipulation scheme. As a researcher, short of actually implementing such a structure in real life and breaking myriad security laws, information-based manipulation is harder to study because the world models required are substantially more complex than those of a limit order book alone. Similarly emergent *action based* manipulative algorithms are feasible if say a large commodity trading company with market power and physical storage capacity integrated an inventory control algorithm with a trading algorithm in a scheme similar to the one described by Stevens and Zhang (2016).

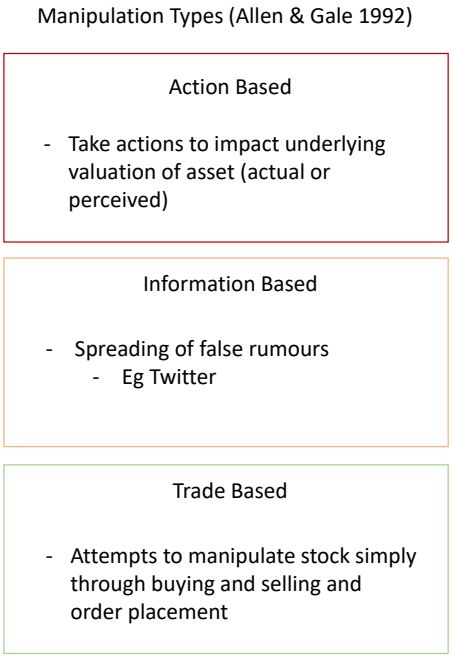


Figure 1.1: Classification of price manipulation activity, adapted from Allen and Gale (1992)

The study of price manipulative strategies within financial markets has a long history. One of the early papers to consider the problem within a rigorous mathematical model is Hart (1977), which proves that uninformed speculators are able to profitably manipulate the price of an asset. Jarrow (1992) builds on this further by defining basic conditions for the existence or non-existence of profitable price manipulation strategies. He finds that if the pricing process only depends on an agent's current holding and not the sequence of trades

used to attain it, then there can be no profitable manipulation. This is the beginning of a noticeable theme in the literature surrounding price manipulative behaviour: proof of non-existence. This is motivated by Jarrow's observation that 'no price manipulation' is analogous to the concept of 'no price arbitrage' which is an important assumption in other areas of quantitative finance such as option pricing (see for example Ross (2004)). In many subsequent market models, the possibility of profitable market manipulation strategies existing is an inconvenience when considering optimisation problems because they are potentially infinitely profitable. This messes up optimisation procedures for obvious reasons. Thus a significant amount of research effort has been undertaken to build trading models where price manipulation is provably irrational - (Alfonsi and Acevedo, 2014a; Donier, 2012; Fruth et al., 2014).

Whilst the papers in the previous paragraph tackle market manipulation by attempting to model the market and its response to orders, an equally valid approach more in line with modern ideas of machine learning would be to try and learn facts from historical data. This data driven approach precludes the training of market manipulative strategies and is principally focused on the problem of detection. This is because any trade-based manipulation relies on the market reacting predictably to order placement and the strategy being able to benefit from the predictability of that response. Methods using historical data are naturally tied to a single realisation of a price series or a limit order book. Here a trading algorithm that learns by doing, cannot learn about the effect of its actions on a market's state, because that trajectory is fixed. Historical data can be usefully used to study market manipulation in the context of classification tasks, i.e., given some historical trading data which may or not be labelled as manipulative, can we construct a classifier to determine whether this unseen trading data is manipulative? This constitutes a very large part of machine learning research regarding market manipulation; Zulkifley et al. (2021) provide a recent survey of approaches. Many methods have been used including Hidden Markov Models (Cao et al., 2015), recurrent neural networks

(Wang et al., 2019) and SVM (Öğüt et al., 2009).

Supervised learning applied to criminal activities is often made harder by imbalanced datasets and the likelihood that not all criminal behaviour is labelled correctly (false negatives). Unsupervised approaches can sidestep this issue to some extent: Leangarun et al. (2019) use the discriminator trained in a GAN (Generative Adversarial Network - see for example Creswell et al. (2018)) to detect price manipulation. Other unsupervised approaches used include clustering through kernel density estimation (Abbas et al. (2018)).

The study of emergent price manipulative trading strategies requires a third course which in some respects sits between the strict model and the model free, data centric approaches. This approach requires a market environment which obeys certain mechanical rules (like a functioning Limit Order Book) and reacts to orders submitted to it. This 'reaction' could be according to a parametric function which has separately been fitted from data. Alternatively, the 'reaction' could be generated by a multi agent simulation, where multiple actors, each interacting with the market using simple rules, on aggregate create a market place which mirrors properties of real markets. Such an approach Cliff and Bruten (1997); Palit et al. (2012)) has been able to create some of stylised properties of markets as described in Cont (2001).

1.3 Research Questions

This thesis will attempt to answer the following questions:

- Does price manipulative behaviour (spoofing) emerge in continuous double auction market where trading algorithms are self-taught (through reinforcement learning)?
- Motivated by the definition of spoofing as the intent to cancel an order at the point of its placement, how can the legal concept of *intent* be established in an algorithm?

- Are lay people willing to accept the concept of intent in algorithmic actors and are they able to interpret evidence for it based on a given definition? This is motivated the observation that criminal trials are decided by juries of lay people (in common law jurisdictions).
- How can we ensure that an algorithmic trader, taught through reinforcement learning never places an order with the intent to cancel it?

1.4 Scientific Contribution

This thesis makes the following contributions to science:

- Demonstrate that spoofing behaviour is a consequence of reinforcement learning and therefore foreseeable.
- Present a definition of intent for algorithmic actors based on current (common) law.
- Conducts quantitative research on the willingness and ability of lay people to judge intent in algorithmic actors and how it differs from their judgement of human actors.
- Demonstrate methods to constrain the behaviour of autonomous algorithms to be law abiding.
- Further widen research and debate in the nascent area of algorithmic crime and regulation.

1.5 Outline of Thesis

In the Chapter 2 we present a literature review covering the subjects of Market abuse and Algorithms (Machine Learning or otherwise)

In Chapter 3 we introduce the limit order book platform **BUCLSE** and the simulation environment built around it which underpins the quantitative research undertaken in this doctoral thesis. Included in this chapter is a trader bestiary which describes the workings of various zero intelligence traders included as

standard with the platform.

Chapter 4 takes the BUCLSE platform, populated with a range of standard and novel traders, and shows how a trading algorithm trained using a variety of Reinforcement Learning methods centred around Q-learning, will readily learn strategies that are price manipulative. This is done by comparing strategies found with an unrestricted action space which will permit order book manipulation and an action space which will not. Analysis of the found strategies is achieved by looking at the frequency of actions chosen, the circumstances under which trading episodes finish and finally by supervised training of a tree classifier using the state, action data gathered during strategy testing.

Chapter 5 defines the concept of *intent* for algorithms by referring to the legal definition of intent that can be found in most common law countries.

Chapter 6 tests a definition of direct intent on lay-people. This has two purposes. Firstly, it is to test whether a common definition of intent differs from people's natural understanding of the concept. Secondly it is to see whether people differentiate between human and AI actors when presented evidence about the intent of the actor.

Chapter 7 takes the US definition of spoofing - placement of orders with the intent to cancel - and trains a RL agent not to spoof in a minimal queuing environment analogous to a limit order book. Two definitions of intent are considered. A statistical definition of intent is rejected in favour of one based on a counterfactual evaluation of disappointment, justified by legal reasoning. A structure known as a shield or ethical governor is used in the training process.

1.5.0.1 Reading Order

Whilst it might be best to read this thesis in the chapter order presented (and the order reflects the development of ideas that I have gone through), it is not a requirement. Chapter 3 introduces the LOB simulation environment that

Chapter 4 uses to train a RL agent so they should be read in conjunction with each other. Otherwise Chapters 5, 6 and 7 are self-explanatory and can be read in any order.

Chapter 2

Literature Review

A literature review concerning Spoofing and related market abuse practices within the Limit Order book (LOB). I find little research concerning market abuse emergence in interactive LOB. Instead I find a concentration of studies on detection methods.

2.1 Introduction

In this chapter I review the existing literature concerning spoofing and related market manipulative techniques. The scope of the search is primarily focused on quantitative research originating from computer science and related subject areas. Subsequent chapters will consider the output from legal research.

Literature concerning spoofing is somewhat limited for many technical reasons which I will discuss later but also an important social/regulatory one. The practice was as MacKenzie (2022) observes, just considered good trading 30 years ago when humans engaged in it, and criminalisation (with enthusiastic enforcement) is a relatively recent occurrence. The change as Mackenzie points out may be connected to the rise and dominance of electronic order books and algorithmic trading. The $920 million fine of JP Morgan Chase in 2020 (CFTC, 2020) for spoofing in precious metals and treasury markets is the largest single

fine for the practice and indicative of the increasing focus that the regulator has on the practice. According to the CFTC's annual enforcement report (CFTC, 2021), the number of cases filed for manipulative conduct or spoofing between 2011 and 2014 averaged five per year; in the three years from 2018 to 2020 there were 19 per year.

Research concerning spoofing can be classified by objective. The largest part of research concerns detection efforts. Related are case studies which analyse certain episodes in market history. A more theoretical strand considers the existence or non-existence of order-based manipulation in parametric models of a market. Often this is motivated by the study of order impact, and the arbitrage problems that spoofing raises if achievable in the model. A final strand of research considers simulations of markets, which are often agent-based models. The objective of these studies might be to consider the profitability and welfare effects of spoofing. A model of the market also allows analysis of possible mechanisms to reduce spoofing. Most pertinently for this thesis, some studies consider the emergence of spoofing strategies through machine learning techniques.

Choosing to classify research by its objective also aligns with the type of data that these studies use. Classification research will predominantly use historic data whilst General equilibrium approaches will use data, usually generated by multi agent simulations.

2.1.1 Empirical research

Spoofing research is often concerned with the classification of spoofing behaviour when presented with historic data. A difficulty in such a task is that labelled data is both rare because exchange data is anonymised; typically the regulator or exchange is the only party in possession of data that identifies individual traders with their orders. The task is further complicated by the likely presence of false negatives in any database (assuming regulators are not

100% efficient in detecting spoofing). An alternative to using labelled data is to identify 'likely' cases of spoofing, which is what Lee et al. (2013) and Mendonça and De Genaro (2020) do in Korea and Brazil respectively. Using data from a brokerage, they reconstruct the order book over the day and identify situations where a party has positions on both side of the order book simultaneously, one substantially larger than the average for the day after, which the large order is cancelled. Lee et al find that traders who spoofed enjoyed higher returns and that stocks which were targeted tended to be smaller in market capitalisation and have higher volatility. Kong and Wang (2014) are able to use a labelled dataset by examining the trading data of someone who was prosecuted for spoofing in the Shanghai and Shenzhen stock exchange. They hypothesise that spoofing is more likely to work when the spoofer can pose as an informed trader, and this is easier when news-flow for the stock is higher. They find that investor sensitivity to orderbook imbalance, that is to say the overweight of one side of the LOB over the other is reduced following a period of manipulation. This they believe indicates that investors learn not to trust the statistic as having informational content after it has been deliberately manipulated by a spoofer.

Tao et al. (2022) present an approach to detect spoofing based on the observation that for a spoofing strategy to be profitable, it must result in a transaction, after which there is an order cancellation. This should imply that in the case that any particular trade is part of a spoofing scheme, the orderbook imbalance distribution pre and post that trade will be different whilst for a legitimate trade it should be similar. They use the Wasserstein distance to measure the distributional difference and calibrate it using level 2 data from the TMX, a Canadian stock exchange.

As Tao et al recognise, a complicating factor to empirical work concerning spoofing is the possibility that it is conducted on multiple exchanges concurrently. The correlation between trade execution and order cancellation in other

exchanges is shown by van Kervel (2015). Practically speaking, this means that orderbook data needs to be aggregated across different exchanges in order to gain a cohesive picture of the situation. Efforts at any single exchange to detect spoofing are therefore hindered and in some cases, it may only be the regulator who has the ability to coordinate such a data collection effort. An even more elaborate problem is the possibility of spoofing across several assets simultaneously. Stenfors et al. (2022) investigate the problem in Over the counter (OTC) foreign exchange currency pairings. They find that predictable moves in one currency pair (EURJPY) can be initiated by spoofing in a different currency pair (EURUSD and USDJPY). They argue that predictability and therefore profitability indicates at the very least a danger that someone will exploit it. In an earlier empirical study, Stenfors and Susai (2021) show the practice of spoofing takes place to various degrees in OTC FX markets. This study also considers the practice of 'pinging', the practice of flashing orders on an order book in order to gain information about the conditions in the market-place. Whilst always considered a dubious practice (Scopino, 2014), as the CFTC makes clear in their guidance note (CFTC, 2013), spoofing is not defined by a motivation to make profit and traders fishing for information will be prosecuted. This position was exemplified in CFTC (2018a) with a $250,000 fine issued to the offending bank.

A common issue with many empirical approaches to detecting spoofing is that they typically require the spoofer to be engaging in the practice for profit, i.e. placing and executing trades as well as placing orders that they intend to execute. As the CFTC make clear, spoofing is not defined by motivation. Quote stuffing is the practice of overwhelming an exchange with orders. It can slow other participants' access to execution and can slow price evolution which might open arbitrage opportunities when assets are also traded at other exchanges (Dalko and Wang, 2020). Egginton et al. (2016) find quote stuffing to have occurred in 74% of exchange listed securities in 2010. Mavroudis (2019) classifies this mode of manipulation as a physical attack on the workings

of the exchange requiring a knowledge of its systems. It is an open question under which circumstances an algorithm could learn a quote stuffing strategy. Perhaps it would be possible from historical data alone if an association could be learned between large order submission episodes and the arbitrage opportunity this might open up. Otherwise, a very realistic simulation of multiple exchanges would be required for the strategy to emerge.

2.1.2 Existence or Non-existence

A surprising amount of research on the subject of order-based price manipulation concerns its existence or non-existence in parametric models of asset markets (see for example Alfonsi and Acevedo (2014b)). As discussed, the evidence for spoofing was relatively scant until recently (Lee et al., 2013) so a market model with no spoofing possible was not obviously unrealistic. The main objective for this avenue of research is connected to market-impact and efficient execution. Market models where order-based manipulation is profitable risk producing arbitrarily profitable liquidation strategies (Klöck et al., 2017). Cartea et al. (2020) consider the optimum execution problem in a parameterised trading environment calibrated with Nasdaq data. Their market model does admit spoofing, but it is limited by a penalty for engaging in the practice which corresponds to the fine for being caught weighted by the probability of detection. They find lower fines lead to more spoofing.

2.1.3 Profitability, Emergence and Welfare of spoofing in market models

Research concerning the emergence of market manipulative strategies such as spoofing is limited. It requires both a simulated environment capable of responding to different strategies and a trading agent who is capable of learning new trading strategies. The risk of emergence, and the difficulty in prosecuting it is predicted, to various degrees in legal research by Scopino (2015); Lin

(2016); Bathaee (2018). Perhaps this is an example of the public imagination of the capabilities of AI in trading running ahead of the reality.

Martínez-Miranda et al. (2016) seek to answer the question: Under what conditions do agents undertake market manipulative behaviour, and what can be done to prevent it? Their approach is one of reinforcement learning, with agents seeking to navigate an obstacle in a maze block world as an analogy for market manipulation (spoofing and pinging). The obstacle in the maze world represents a lack of liquidity at a price level that the agent wants to buy at. It is 'moved' by the agent undertaking a manipulative action. The authors consider two mechanisms for discouraging the manipulative behaviour which otherwise emerges from their setup. Firstly, they increase the costs of manipulation to the agent (fines). Secondly they model the exchange led practice of 'controlled liquidity' by adding uncertainty in the transition probabilities of the model. Spoofing creates an orderbook imbalance for the purpose of giving a perception of skewed supply or demand. Controlled liquidity is a possible response from exchanges where large imbalances are automatically rebalanced in some prearranged mechanism. Thus, in an exchange that uses controlled liquidity mechanisms, manipulative action will only work with a certain probability.

The idea of representing a profit maximisation problem as one of two dimensional movement in a maze action space is novel but is quite an abstraction. As the authors admit, the results might not be robust: "other grids with more complex structures may also reproduce trading strategies, but the manipulative behaviour may not emerge as an optimal control according to the simulated market conditions, thus eliminating the core of the analysis we present".

Mizuta (2020b) considers the question of emergence of market manipulative trading strategies in an agent-based model. The agent-based model is described in Mizuta (2020a) and recreates a fully functioning double auction market. Zero intelligence type traders take turns to buy or sell single units of an asset according to a strategy which mixes fundamental valuation and

momentum-based criteria. The learning mechanism of the agent is evolutionary, with genes expressing unconditional buy, sell or pass action decisions. All strategies are assessed against the same market, so that an identical strategy would lead to an identical profit. Multiple learning agents with randomised genes are assessed and a random crossover and mutation process evolves the best performing strategies of a 'gene pool'. Mizuta contrast strategies in an environment without market impact to those obtained in full simulation. In the case of the former, a fundamental type strategy emerges where the strategy buys when the asset is historically undervalued. In contrast, in the fully simulated environment, the evolved strategies exhibit overbuying or overselling, which drives the market upwards or downwards to extreme price levels, after which the trader exits their position. This Mizuta claims, is indicative of a manipulative pump and dump type strategy. The experiment neatly demonstrates the different strategies which emerge in otherwise identical environments depending on whether market impact exists or not, but the fact that the learned strategies are not conditional on market conditions weakens the strength of the conclusion.

Whilst not a proof of emergence, Withanawasam et al. (2013) show in some conditions a pump and dump strategy is profitable in a multi agent simulation of a LOB market. Profitability is dependent on the presence and concentration of 'technical traders' who use recent data in order to make trading decisions. Manipulating agents are able to 'ignite' strong periods of price momentum by buying shares in an initial period, waiting for the technical traders to bid up the asset price in a momentum period before exiting their positions in cooling off period during which the price of the traded asset collapses. The authors extend the LOB model of Maslov (2000) by adding a class of technical traders to the existing liquidity traders. At each period a trader type is drawn at random according to a constant distribution parameterised by (p_L, p_T, p_M), the proportions of Liquidity, Technical and Manipulating traders respectively. A trader submits a buy or sell order with probability μ, $1-\mu$. They choose a

limit order with probability θ or a market order with $1-\theta$. These parameters vary according to trader type. Market orders are offset from the contraside best price by a random integer $x \in [1, 2, 3, 4]$. This price choice was found in Withanawasam et al. (2010) to better match price movements observed in real markets than Maslov's original specification. Technical traders make their buy or sell decisions based on the previous period's observed price movement, order type (buy or sell, limit or market) and whether they think that order's originator was 'informed' or uninformed. They (falsely) believe informed traders will only buy (sell) if the price rose (fell) in the last period. The authors find that the profitability of the manipulating strategy increases with p_T, the proportion of technical traders in the environment.

In Wang and Wellman (2017); Wang et al. (2020) and in greater depth Wang (2021), a similar dichotomy of non-manipulable and manipulable agents are used in an agent-based model to investigate the practice of spoofing in a LOB. The objective of the experiment is to assess the welfare implications of spoofing traders being present in a market.

The non-manipulable agent class are a modified type of the 'Shaver' zero intelligence trader described by Cliff (2018). The manipulable agent class is called a Heuristic Belief Trader (HBL). An agent receives a noisy signal (Gaussian additive unbiased noise with variance σ_n^2) of the fundamental value every time it trades. The Shaver traders just submit orders with a random sized offset to their valuation of the asset. The HBL trader users order book information to assess the probability of execution at each price and then chooses a price to maximise their utility. This is also dependent on an agent specific parameter which corresponds to the marginal utility for acquiring another unit of the asset.

The spoofing strategy consists of an agent placing a large quantity order one tick behind the best bid. Since the other 'background' traders can only trade single units, the spoofer is able to avoid any risk of execution - they are assumed

to have a privileged position where they can cancel and replace their order if the best bid/ask moves. In some of the experiments the spoofer is given an exploitation strategy in order to assess their profitability. The spoofer will buy when there is an order below estimated fundamental mean, it will then begin spoofing as described above and subsequently sell if a limit order bid is greater than estimated fundamental mean.

Parameters for the traders are found using Empirical Game-Theoretic Analysis (EGTA) which is an iterative procedure to find Nash-equilibria amongst parameterised strategies. The surplus of the equilibrium strategies are then compared with and without the presence of a spoofer. They find that the presence of HBL traders (i.e. traders who use order book information) increases aggregate surplus and the presence of a spoofer reduces it.

Liu et al. (2022) use EGTA to investigate whether frequent call markets are also vulnerable to spoofing. Frequent call markets (Wah et al., 2015) operate as a sequence of closed bid auctions over a timeline discretised by clearing periods. Orders submitted during any clearing period are not time prioritised. They were developed as a putative solution to the latency battle that High Frequency Traders (HFTs) wage in order to enable, what some would consider, market manipulative spoofing strategies (Cooper et al., 2016; Dalko and Wang, 2020). Liu et al use a similar setup and method to Wang and Wellman (2017). They find spoofers have a similar negative effect on overall market efficiency but a smaller one than one found in a traditional continuous double auction market. They also find that the profitability of the spoofer is dependent on their speed in updating their orders to avoid unexpected execution.

2.2 Justification of research problem based on literature review

Research concerning spoofing can be classified by objective. The largest part of research concerns detection efforts. Related are case-studies which analyse particular episodes in market history. A more theoretical strand considers the existence or non-existence of order based manipulation in parametric models of a market. Often this is motivated by the study of order impact, and the arbitrage problems that spoofing raises if achievable in the model. A final strand of research considers simulations of markets, which are often agent based models. The objective of these studies might be to consider the profitability and welfare effects of spoofing. A model of the market also allows analysis of possible mechanisms to reduce spoofing. Most pertinently for this thesis, some studies consider the emergence of spoofing strategies through machine learning techniques.

There exists a gap in existing knowledge surrounding the genuine *emergence* of market abusive behaviour within a continuous double auction market from a learning trading algorithm. By extension, there is even less research concerning the problem of an algorithm learning under a restriction not to commit market abusive behaviour. Whilst Wang and Wellman (2017), Withanawasam et al. (2013) do prove the viability and stability of such behaviour in a realistic limit order book environment, their manipulative agents behave with an experimenter imposed behaviours. Martínez-Miranda et al. (2016) goes part of the way to study the question of market abuse as learned behaviour, but the approach is quite abstract to be satisfactory.

For market abusive behaviour to successfully emerge I believe there to be three necessary conditions to be satisfied.

1. Price manipulation requires actions taken in the current period to affect the state of the market in the future. Price formation must be endoge-

nous to the model.

2. Price manipulation is concerned with the deception of other traders, so the behaviour of other traders (the marks) must be accounted for in a model of the situation.
3. The trading strategies of the other agents must be conditional on the state of the system in a way which can be affected by manipulator as observed in Wang and Wellman (2017) and Withanawasam et al. (2013).

Chapter 3

Platform: BUCLSE

I present a Limit Order Book simulator named BUCLSE, capable of multiagent simulations, offering flexibility over underlying supply and demand setups. The environment has been designed specifically for the study of market manipulation and market abuse. It is programmed in Python and has been designed for future extension to a large multi-core setting.

3.1 Introduction

The study of emergent manipulative trading behaviour necessitates an environment which provides feedback to the trading agents within it. That is to say, the actions of the every trading agent, have the potential to affect the environment and thereby the actions of the other agents. This precludes the use of historic pricing data as a way to explore the problem. In machine learning terminology, learning must be done predominantly online (as apposed to a batch setting).

This is a two edged sword. On one hand data can be obtained for free; it is generated by the trading environment itself. Any enquiries about the cost of acquiring multi-level orderbook trading data from external vendors will convince the reader of the attraction of this. It is kept under lock and key by

a small number of guardians and the cost of the key is prohibitive for most independent researchers[1]. On the other hand there are two problems with a market simulation approach, one theoretical and one mechanical. Firstly the theoretical problem is that there is no guarantee that any data produced in a market simulation is realistic and of any practical use. Thankfully, since the objective of this research is not to find magical money making machines but to investigate the emergence of illegal behaviour and how to arrest it, we don't need to worry too much about this problem. A robust finding of emergence should mean that the problem of emergent spoofing strategies is foreseeable and foreseeability does not make huge demands about likelihood thus the realism of the simulator is not of primary importance. That said, the trading environment will work as realistically as possible and is designed to have the flexibility to potentially produce a realistic simulation. The parameter search to achieve this is left for another research project - methods themselves based on machine learning techniques to do this are emerging (Bai et al., 2021). Previous work on multi agent models and zero intelligence traders has consistently shown that certain market 'facts' are reproduced quite robustly as an artefact of the price matching model found in limit order books. (Cliff and Bruten, 1997; Palit et al., 2012; Gode and Sunder, 1993). To reiterate, it should be enough to demonstrate that if a certain market manipulative behaviour emerges in a limit order book simulator, the emergence of market manipulative behaviour is foreseeable in more complex simulations and real life. This is because the concept of foreseeability of an outcome does not rely on proving a high likelihood of that outcome. Anyone therefore working on auto-didactic trading algorithms should be very aware of the risk of market manipulative trading strategies arising either from simulation or learning in deployment.

The mechanical problem relates to the efficiency of the trading environment.

[1] An *academic* service LOBSTER providing limit order book reconstruction data for example is €5000 pa. `https://lobsterdata.com/`

If data generation and learning are online, can we make the environment sufficiently efficient enough to generate data at the speed and volume which modern machine learning methods require? Benchmark reinforcement learning problems enjoy fast, lightweight toy environments as a given. For a market simulator to be suitable for reinforcement learning, it should be suitably fast and robust, and that requires a significant amount of additional effort. At the same time, a balance has to be made between speed and code interpretability.

3.2 Background

BUCLSE was created as a means to an end - the study of emergent market manipulative behaviour via reinforcement learning - because no other solution existed with the capacity to do what I needed. Since I created it, other open source platforms have become available such as ABIDES (Byrd et al., 2019; Byrd, 2019), SHIFT (Alves et al., 2020b) and MAXE (Belcak et al., 2020). Aside from indicating a preponderance for capitalisation amongst programmers, the growing interest in Agent Based Modelling (ABM) applied to finance reflects the increased application of reinforcement learning to trading problems (surveyed in Meng and Khushi (2019); Pricope (2021); Borrageiro et al. (2022)) and the realisation that static data cannot service this method properly. Of the simulation environments mentioned above, perhaps ABIDES is best supported; just as with BUCLSE it is integrated with common reinforcement learning package Pygym by Amrouni et al. (2021). It has been used to consider optimal market making problems by Gašperov et al. (2021) and in multi agent reinforcement learning by Karpe et al. (2020).

3.3 Introduction to BUCLSE

BUCLSE was built upwards from BSE (Cliff, 2018) to provide a minimal trading environment to investigate the emergence of illegal trading behaviour. BSE was built in Python 2 as a teaching aid for a university algorithmic trading

course. As such it was a convenient starting point for BUCLSE. BUCLSE is programmed in Python 3 and suitable for versions 3.7 and beyond.

Within a multi agent trading environment there are four main elements to consider:

1. The **Traders** - Variable in quantity and type, these are the agents that make trading decisions within the simulation.
2. The **Exchange** - This provides the price matching mechanism through which traders transact.
3. **Supply and Demand** dynamics for the asset - This can be described through supply and demand curves, or an underlying fundamental price for an asset. It is the information source that some trader use to reach trading decisions and drive the simulation.
4. The experiment **Controller** - The most varied of the four elements, it is the mechanism that coordinates the previous three elements into a simulation. It is 'outside' the simulation is just directs the elements to work in the way specified by the experimenter.

Since Python is an object oriented language, these elements have natural implementations as objects and appear as such in BUCLSE.

A point of differentiation in BUCLSE is the method through which the elements interact. From inception BUCLSE has been designed with one eye towards multi-core operability with the introduction of timer and messenger objects. As such we add two more necessary elements to the preceding list:

1. A **Message** system - The mechanism through which the various simulation objects communicate with each other.
2. A unified **Timer** - Time needs to be agreed upon between all parties since this is an online, sequential simulator. The passage of time is controlled by the Controller [2].

[2]We considered natural time, but this is problematical when some types of trader are

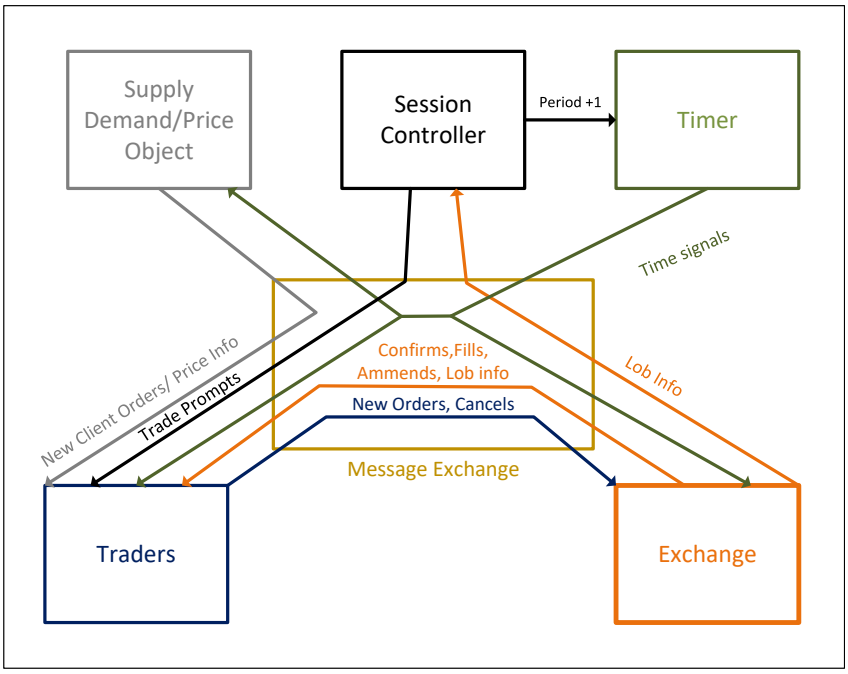


Figure 3.1: A message route map for the BUCLSE system. In a single core setup, the timer object is common to all objects, obviating its need to send messages. As the system is developed for multi core setups and asynchronous operation, time updates via message will become necessary.

3.4 Timer and Messenger

The messenger object is the conduit through which all elements of the simulator communicate with each other. Enforcing inter-object communication through a messenger system is desirable because it removes the limiting factor of a single computer's memory or CPU on the simulation. Thus BUCLSE has the potential to be scaled to an arbitrary number of independent cores - a multi agent simulator using multiple computing units. A route map of the system's messages is shown in figure 3.1 on page 53.

The messenger object is in effect a message server which maintains a list of subscribers. Traders subscribe and submit messages to the messenger which in turn sends them on to the exchange and vice versa. Likewise the Controller can coordinate its traders through the emission of messages. Thus, every element of

able to reach decisions faster than others. The current structure does not preclude it in future implementions

the BUCLSE becomes independent of every other element by having a method of receiving and sending messages. Whilst in the single core environment this appears to be an over elaborate way of getting two objects within the same memory to communicate with each other, the log of passed messages becomes in itself a very useful debugging tool. [3]

The timer object is a simple object which is common across all elements within BUCLSE [4] It allows all objects in the environment to agree about what 'time' it is within the simulation. Unlike using a simply clock however, we are able to control the passage of time within the environment. This means differing computation time for agents with different strategies is not an issue even if the simulation environment is distributed across multiple cores.

The object based approach and the use of a messages to communicate between them can be seen as an example of agent based programming as introduced in Shoham (1993).

3.5 Exchange

The Exchange object maintains a Limit Order Book (henceforth LOB) for a single asset. Individual orders at a price are given a price based on First In First Out (FIFO ordering). Orders can be of any integer quantity.

For ease of communication, I created an Order object which contains the vital information of an order and is used extensively in the communication between trader and exchange. The exchange can receive two types of message, "New Order" and "Cancel". It can send "Fill", "Ammend"[5] and "Confirm" messages. Reject messages were not necessary since the trading agent in any period has up to date information and no agents can place orders before them.

[3] And as the author found out, multi agent simulations are finickity to develop.

[4] Time changes can be communicated and coordinated through messaging in the true decentralised implementation

[5] Orders with heterogenous quantity necessitate the ability to ammend orders on the orderbook since orders can be 'partially filled.'

In the event of a receiving a New Order message from a trader, the exchange replies with a quote ID and then checks for execution if the order price crosses best bid or ask. On execution, all counterparties are contacted with Fill messages (and Ammends when applicable). Market orders can be submitted if the limit order price is better than the current best level.

The exchange also functions as an information repository for the traders. It can respond to the following types of request:

1. **LOB requests**: Publish a version of the LOB which is anonymised. Best Ask and Best Bid are also given.
2. **Tape requests**: Publish the history of new orders, cancels, fills and trade ammendments up to a user defined duration.
3. **Personalised position**: The positions (levels and ordering) of a trader within the current LOB.

In future versions, it will provide more advanced order and market statistics so as to prevent trading agents replicating operations.

3.6 Supply and Demand

In some ways this is the hardest of the main four elements to describe, but all multi agent market simulations require some kind of mechanism which drives the moves of the LOB. More often than not in ABM approaches to LOB simulation there is just an underlying 'fundamental' price sequence which we assume implicitly reflects supply and demand conditions.

The fundamental price setup is the one which I use in the following chapter so the one I will describe here but BUCLSE works with an alternative formulation, described in Appendix A. This is where customer order are distributed to traders, drawn at uniform from a supply or demand curve of prices. These curves can be shifted over time to alter the equilibrium price of the market.

3.6.1 Supply Demand setup: Fundamental Price

This formulation specifies a fundamental price sequence from which traders receive noisy signals. The price process specified in Wang and Wellman (2017) and shown in Equation 3.1, is a mean reverting random walk and noise is Gaussian. As long as the traders apply the correct bayesian update regarding their beliefs concerning the price process, there is no particular reason why this price process couldn't be different.

$$r_t = max\{0, \kappa\bar{r} + (1-\kappa)r_{t-1} + u_t\} \quad \text{for } u_t \sim N(0, \sigma_s^2),\ \kappa \in (0,1) \tag{3.1}$$

As before, a sequence is set on environment initiation which determines when traders receive information and are subsequently prompted for orders. At most one trader is prompted for an order per period. The price, noise and trader prompt sequences are set before the experiment begins, allowing the repeat of any experiment.

3.7 Traders

The role of the basic trader object is to receive information and submit orders to the exchange.

From the Controller it receives order 'Prompt' messages which invite the trader to submit or refresh orders to the exchange through invocation of the "getOrder" method. In future versions where the traders are run on independent, individual cores, prompt messages may be unnecessary, the trader will decide themselves when to to submit orders. A benefit of 'inviting' traders to submit is that the invitation sequence can be saved to recreate the experiment.

After sending new order or cancel order instructions to the exchange they receive an order confirm message in return. When an existing order is filled, they will receive 'Fill' and 'Amend' messages from the exchange. On receipt

of these messages, internal records are updated accordingly.

A bookkeeping function exists to keep track of profitability once trades have been executed. Various data structures exist within the trader object to keep track of submitted orders and the current state of the LOB. Where a trader has the capability of holding inventory, unrealised profit is calculated on a FIFO basis.

Typically, the market variables of a trader (with which it makes trade decisions) are updated at the end of each period on receipt of a "Respond" message sent by the controller.

Additionally in a setup where there is an underlying fundamental price sequence driving dynamics, the trader can receive private information about the price through prompt messages.

In practice, specific traders are subclasses of the the default trader object. They are typically differentiated by the mechanism through which they decide on orders to submit to the exchange. Depending on what information they use to reach trading decisions, their internal data maintenance methods will also differ.

To save on memory and calculation time, the statistics based on order book information which traders maintain are calculated at a trader class level. This means that only one representative trader of that class need update market statistics each period for all other traders to have the same information. Whilst this decouples the overhead of having more traders of any particular class by eliminating duplication of market statistic calculation, it does come at the expense of restricting decentralisation (Traders of any particular class would have to be housed in the same core). As a downside this makes running parallel experiments more complex.

3.7.1 Trader Bestiary - Fundamental Price type

This type of traders is given noisy signals about an underlying price sequence by the Controller. Traders are free to take inventory within bounds and can submit both bids and asks simultaneously. Typically they are given a randomly generated preference over their inventory holdings which affects their price submissions. An alternative set of zero intelligence traders adapted from Cliff (2018) are described in Appendix A which work in the supply and demand experimental setup.

3.7.1.1 Wang Wellman Zero Intelligence Trader WWZI

WWZI traders receive a noisy signal of the fundamental value $\tilde{r}_t + \sigma_s.\epsilon$, with $\epsilon \sim \mathcal{N}(0,1)$ every time they are due to submit orders. WWZI traders are free to submit both bids and asks should their future inventory on execution of that order not exceed their inventory limits. WWZI are Bayesian and estimate a posterior value of the future[6] fundamental value of the asset based on their new signal and a posterior summarising previous signals. They are assumed to know the parameters of the fundamental price process and their signal.

In the case of buying they add their buy preference Θ^{q+1} (which is a function of their inventory q) to their asset valuation $\hat{r}_{t+F}$ and choose a bid price uniformly at random from the range $[\hat{r}_{t+F} + \Theta^{q+1} - K, \hat{r}_{t+F} + \Theta^{q+1}]$ for some constant $K \in (0, \infty]$. Selling is similarly defined. See also exposition on page 77.

3.7.1.2 Heuristic Belief Traders HBL

HBL traders Wang and Wellman (2017) are as WWZI traders in their estimate of a future fundamental price $\hat{r}_t(F)$. In addition they estimate the probability of order execution as a function of order price p. They then choose a price p

[6] In the original formulation of Wang et al. (2018), the traders estimate the final fundamental value of the asset. We found that this caused agents to choose $\bar{r} = 100$ as their valuation in all but final most periods. We therefore made the traders choose a valuation $K = 10$ periods into the future

which maximises their expected surplus $(\hat{r}_{t+F} - p) f_t(p)$.

The probability distribution is given by:

$$f_t(p) == \begin{cases} \frac{TBL_t(p)+AL_t(p)}{TBL_t(p)+AL_t(p)+RBG_t(p)} & \text{if buying} \\ \frac{TAG_t(p)+BG_t(p)}{TAG_t(p)+BG_t(P)+RAL_t(p)} & \text{if selling} \end{cases} \tag{3.2}$$

The three and two letter quantities of the type $XYZ(p)$ and $YZ(p)$ in equation 3.2 correspond to the logical intersection of certain types of orders that have been submitted to the exchange. Specifically XYZ is shorthand for $Q(X \cap Y \cap Z(p))$ where X, Y, Z are defined as follows:

$$X \in \begin{cases} T & \text{transacted orders} \\ R & \text{'Rejected' orders} \end{cases} \tag{3.3}$$

$$Y \in \begin{cases} A & \text{Asks (sells)} \\ B & \text{Bids (buys)} \end{cases} \tag{3.4}$$

$$Z \in \begin{cases} L(p) & \text{\# orders less than or equal to price p} \\ G(p) & \text{\#orders greater than or equal to price p} \end{cases} \tag{3.5}$$

and $Q(O)$ is the quantity in order set O. Trade history is restricted to the last $L = 100$ periods.

The formulation of equation 3.2 originates from Gjerstad (2007) but their concept of order rejection was replaced by Wang and Wellman (2017) through a decay factor analogous to order rejection: Orders are considered rejected if in the LOB for longer than a period termed the 'grace period'. Else their quantity is weighted by their age as a fraction of the grace period. We used a grace period of 20 periods.

3.7.1.3 Noise Traders NOI

The Noise trader receives no private signal of the underlying fundamental price sequence. Instead their estimate of the underlying fundamental price $\hat{r}_{t+F}$ is a quantity weighted average price of all recent (within memory) bids (asks) submitted to the exchange less any recent cancelled bids (asks). As before this is modified additively by an inventory preference Θ^{q+1}. Because this formulation would rarely lead the Noise trader to ever improve best bid or ask, the trader also adds (subtracts) a constant to their bid (ask) price when the most recent orderbook shows an increase (decrease) in best bid (ask). For our experiments we chose a memory of 20 periods and the constant equal to one.

3.7.1.4 Imbalance Traders IMB

This class of zero intelligence traders places bids when a statistic called Order Flow Imbalance (OFI(K)) is greater than zero and places ask orders when the statistic is negative. In the bid case their choice of order price for trader i is given in equation 3.6. Order price for ask orders is determined analogously.

$$p_{t,i}^{CONT} = p_t^{B(1)} + c_i.OFI_t + \Theta^{q+1}$$
$$\text{for best bid} = p_t^{B(1)} \quad \text{and for } c_i \sim U[0.2, 0.8] \quad (3.6)$$

The Order Flow Imbalance statistic originates from Cont et al. (2013) and is defined in equations 3.7 and 3.8 for trader i. Here $p_s^{B(1)}$ and $q_s^{B(1)}$ mean best bid price and best bid quantity respectively at time s. Similarly $p_s^{A(1)}$ and $q_s^{A(1)}$ for best ask.

$$OFI(K) = \sum_{s=t-K}^{t} e_s \quad \text{where} \quad (3.7)$$

$$e_s := \mathbb{1}_{(p_s^{B(1)} \geq p_{s-1}^{B(1)})} . q_s^{B(1)} + \mathbb{1}_{(p_s^{B(1)} \leq p_{s-1}^{B(1)})} . q_{s-1}^{B(1)} - \mathbb{1}_{(p_s^{A(1)} \leq p_{s-1}^{A(1)})} . q_s^{A(1)} + \mathbb{1}_{(p_s^{A(1)} \geq p_{s-1}^{A(1)})} . q_{s-1}^{A(1)} \quad (3.8)$$

The coefficient $c_i \sim U[0.2, 0.8]$ is drawn at random when trader i is spawned and is approximately equal to the least squares coefficient β obtained when fitting the equation $\Delta p_t = \alpha + \beta OFI_t(K) + \epsilon$ in the environment without the IMB traders. Similarly the memory coefficient K_i is drawn at random for each trader on spawning with distribution $K_i \sim U[5, 15]$.

3.8 Market Session Objects

The final element organises the experiment - it defines traders within the market, and the supply and demand related schedules and records any information where appropriate. The Market Session is able to control the passage of time via the timer object, thereby controlling the entire experiment. Experiments are structured as a set sequence of events that happen within a period. Periods are repeated by incrementing the timer object until the timer has reached its limit.

On initiation the fundamental price sequence, noisy signals thereof and trader submission sequence are defined. The sequence for a period then proceeds as follows:

1. A trader is picked to submit an order to the exchange. It is given a signal about the value of the fundamental with which to base its order submission. It can submit both bids and asks concurrently as long as it is below its inventory limits.
2. Trader submits order directly to exchange and any resulting trades are processed.
3. Traders update their records according to changes on the LOB.

3.9 Reinforcement Learning adaptation

BUCLSE can be easily adapted for the purposes of reinforcement learning with the addition of a RL Trader object and an RL Environment object which sits above the experiment controller.

The RL trader inherits much of the machinery developed for other traders (principally order submission and bookkeeping). It has an interface to submit any type of order to exchange.

The RL Environment object is a subclass of the Environment class found in OpenAI Gym. This was chosen because of its simplicity and familiarity to anyone who has undertaken RL research. It also means standardised RL toolkits can be brought to BUCLSE at low cost.

The bulk of the effort in adapting the platform for RL is spent formatting state space information (and to a lesser extent action space and writing reward functions). Since this is experiment specific, the RL Environment object will always be quite variable and adaption will be up to the user. As a minimum requirement custom Step and Reset methods should be written. The Step method takes the user's action as input, applies it to the trading environment and returns the resulting new state. The Reset method is invoked when a training episode is deemed to have ended. It resets the environment back to starting conditions for a new training episode and returns an initial state for the RL trader. Examples of the Reset and Step methods for Experiment 1 are described in detail on pages 83 and 83.

3.10 Future development

We would like to adapt the exchange to accept FIX protocol message. Financial Information eXchange or FIX, is the method through which most exchanges receive and communicate trade instructions with participants. Making BU-

CLSE compatible with this standard would allow any algorithm developed for commercial deployment work with the simulator.

As a function of this development, the exchange should be developed to accept a wider variety of orders. Time dependent orders (orders with a expiry time) are a commonly used in real markets and would be useful in simulations where traders have a prior over how long their beliefs about the market are valid.

Early tests using the simulator for Reinforcement Learning have shown that the simulator is not fast enough. We think that the simulator can be optimised. This might involve translating it into a higher level language (C++), a faster language (Julia), a hybrid approach (Cython) or an attempt to parallelize the existing exchange.

Visualisation is an important feature for demonstration. In particular live or online visualisation whilst events are unfolding. We would like to incorporate a visualisation server such as Visdom into the simulator.

In initial tests for a multi-core setup, we used the MQTT message protocol because it is lightweight, easy to understand and is freely supported in Python. A further attraction to using a messaging system in the environment is that traders and exchanges communicate with each other using a standardised messaging protocol called FIX (Financial Information eXchange). Further iterations of BUCLSE might adopt this protocol, thereby facilitating the testing of actual trading algorithms within the simulation environment.

When running multi agent simulations we have noticed a duplication of work surrounding the calculation of market based statistics. This slows simulation performance and increases the code complexity of writing zero intelligence traders. We think it would be more efficient to centralise statistic calculation at the exchange level and then distribute from there. This is a reflection of real life, where subscribers to exchanges pay for additional processed data.

Market manipulation succeeds by changing or aligning the beliefs of other market participants Dalko et al. (2020). A benefit from using a multi agent simulation of the market to study market abuse (over a less explicit parametric representation of the market) is that we have the ability to measure the effect of strategies on participant's beliefs. Such analysis is an avenue to be explored in the future.

3.11 Summary

In this chapter we present a limit order book and surrounding simulation environment called BUCLSE which is designed specifically for the task of studying algorithm led market abuse. This objective relaxes the requirement for its output to be fully realistic since its purpose is to determine the foreseeability of market abuse emerging. It has been built around the BSE of Cliff (2018) but extended to allow such things as non-unit quantity trades, multiple orders per trader, FIFO accounting, trader types with different latency characteristics, predefined trader selection sequences, a decentralised timer and a messenger system for objects to communicate with each other. We also show how different supply and demand setups can work with the simulator (as in Wang et al. (2018)). The direction of design has also been towards decentralisation of exchange from test environment and market participants. This is motivated by knowledge of the heavy computational workload that deep learning requires. A simulation platform that is truly decentralised has greater scope for multi agent learning.

Chapter 4

Emergence of spoofing in a RL LOB environment

In this experiment, a Reinforcement Learning (RL) trader is trained in a multi agent simulation of a Limit Order Book (LOB) to develop a profitable selling strategy in a repeated, trading task. The trader is given the option to posting additional bid orders but is penalised if these orders are executed. Posting bid orders will increase the Order Flow Imbalance statistic which one class of trading agent uses to make trading decisions. This can encourage this class of trader to respond by submitting bid orders in anticipation of a higher future price. By comparing the trader's performance with a restricted action space which does not include the posting of bid orders, I find that the trader will learn to manipulate the LOB in its favour.

4.1 Introduction

In this experiment I aim to investigate whether a market abusive behaviour known as spoofing is exhibited by a trading agent which learns via reinforcement learning in a multi agent simulation of a limit order book for a single asset. Spoofing is defined differently across markets:

- In the USA it is defined as "bidding or offering with the intent to cancel the bid or offer before execution" [1].
- In the United Kingdom the definition is wider: Market abuse is committed if the trader commits orders or transaction which "give, or are likely to give, a false or misleading impression as to the supply of, or demand for, or as to the price of, one or more qualifying investments, or secure the price of one or more such investments at an abnormal or artificial level" [2]

In this chapter I will take spoofing to mean the deliberate manipulation of the future state of the limit order book by the strategic submission of orders. The practical benefit of spoofing for a trader in this experiment is to distort asset price upwards or downwards by influencing other participant's perception of supply or demand within the market. Chapter 7 will consider the US definition in a more general sense. Whilst spoofing is typically thought about as an activity to increase trading profits, this motivation is not a prerequisite.

The experiment uses the BUCLSE LOB simulator introduced in the previous chapter, which is an augmented version of the BSE in Cliff (2018). The market is populated with a variety of zero intelligence type traders and heuristic traders as found in Wang and Wellman (2017) along with two original types of heuristic trader. Heuristic traders are a necessary component for spoofing to work - someone in the market has to trade using order book information because this is the vector through which a spoofing strategy is able to deceive them (Withanawasam et al. (2013); Wang et al. (2018)). I assess the presence or absence of spoofing by comparing the Agent's profitability in two scenarios where the agent does have the action space capable to spoof and one where they do not. I will then examine the RL agent's policy function by approximating it through a decision tree trained on the state input and action output

[1]The Dodd-Frank Wall Street Reform and Consumer Protection Act, 7 U.S.C.A. Âğ 6c(a)(5)(C)
[2]UK FSMA 2001 118 5

of the agent's policy obtained during training.

Alongside conventional Reinforcement Learning (RL), the experiment also uses the Dyna-Q (Sutton, 1990) method of learning whereby the RL trader (the Agent) learns a trading strategy both through *direct* interaction with the environment and through *planning* using a self constructed model. The Agent is able to use this model to improve their strategy without having to interact with the environment in a planning stage. This approach was inspired by practical considerations: Whilst the market simulator is reasonably fast, it cannot generate data quick enough to slake the thirst of modern deep RL methods. Market-simulation using a model trained by the Agent is much faster since training episodes can be run without the computational baggage that a multi-agent market simulation requires. Reinforcement Learning is an inherently wasteful process and time-consuming when state and action spaces are anything but small. By learning a model of the environment, I expect that our Agent is more efficient. Moreover the model should aid with generalisation and opens up alternative, model-based planning and learning techniques in future work. I feel that the concept of trading agents learning a simplified model of the world is a natural one and a fair reflection of algorithmic traders in the real world. Learning by doing is an expensive activity to do in real markets, it makes sense for policy exploration to be done in the agent's zero cost simulation [3].

The **direct** stage of the learning process is through deep-Q learning coupled with experience replay as used by Mnih et al. (2015). The attraction of this approach is that previous experience can be retained and reused efficiently. This is possible because Q-learning is an off policy learning process whereby it is assumed that data is not dependent on the learning agent's current policy. The Agent estimates its value function (the discounted value of being in any state and taking any action) using a neural network. Since neural networks

[3]This experiment involves a performing a market simulation where an learning agent learns and operates a market simulation to learn - a story within a story

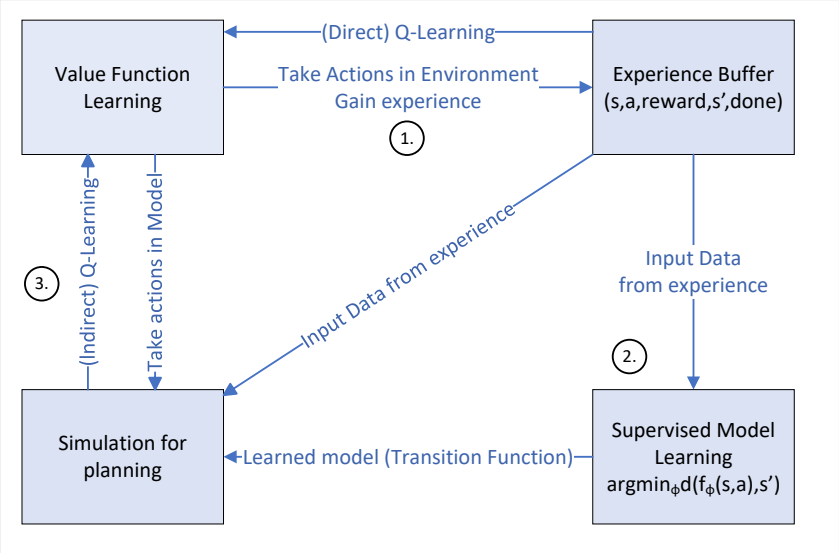


Figure 4.1: A schematic of the Dyna-Q method of learning and planning as adapted from Sutton and Barto (2018). The Numbered stages refer to those parts of the process that require machine-learning both. 1 and 3 are both RL-learning processes, whilst 2 is supervised learning.

can be efficiently evaluated and optimised with batch data, deep-Q learning and experience replay have been shown to work well together.

The **planning** stage of the learning process makes further use of data gathered by the Agent. It is divided into two stages: **Supervised model learning** and **Simulation for planning**. In the model learning stage, the Agent learns a state-transition model through which it can use to simulate the trading environment. This is a mapping of state and action to next state. The training of the Agent's model is through supervised learning, using the data already gathered in the agent's experience replay buffer. The simulation for planning stage proceeds much as the direct learning stage with the agent learning its value function but with the internal model providing state-transitions instead of the actual environment.

4.2 Background

4.2.1 Problem as a MDP

The learning problem of the trader can be modelled as a Markov Decision Problem (MDP) defined by the tuple (S, A, R, P, μ) where:

- S is the set of states. This is divided between public states S^{LOB} and private states specific to the trader S^{Tra}. An example of private state is the trader's level of inventory, or their level of unrealised profit and loss.
- A is the set of actions available to the trader. This can be described as:
  1. Place limit bid (ask) order at price $p_b - k^{min}$ (or $p_a + k^{max}$), for quantity q, $k^{min} = 1, \dots p_b - 1$, $k^{max} = 1, 2, 3 \dots$ where p_b (or p_a) is best bid (ask).
  2. Cancel order (bid or ask)
  3. Liquidate open positions at market.
- $R : S \times A \times S \to \mathbb{R}$ is the reward function. In this experiment it is deterministic and not dependent on the prior state s_t only s_{t+1}. It is a period-wise decomposition of the profit that the trade receives over the episode, described in Equation 4.10.
- $P : S \times A \times S \to [0, 1]$ is the transition probability function of the model. $P(s'|s, a)$ is the probability of transitioning from state s to s' after taking action a. I will denote this $P^a_{s,s'}$
- $\mu : S \to [0, 1]$ is the distribution of the starting state.

The trader seeks to find an optimal stationary policy $\pi \in \Pi$, where Π is the set of all mappings from states to probability distributions over actions $\pi : S \to \mathcal{P}(A)$.

The objective function of the trader is to maximise the expected discounted

sum of rewards from a strategy for some discount rate $\gamma = 0.99 \in (0,1)$:

$$\max_{\pi \in \Pi} J(\pi) := E_{\pi}\Big[\sum_{t=0}^{T} \gamma^t R(s_t, a_t, s_{t+1})\Big] \tag{4.1}$$

The transition function is unknown to us (and the Trader) but we are able to sample from it using the trading environment which I have adapted. The distribution(s) of these samples are dependent on the current policy of the agent. This is something that Q-learning ignores.

4.2.2 Q-learning

The action value function or Q function is the expected value of taking action a in state s before following policy π thereafter.

$$Q_{\pi}(s,a) := E[R_1 + \gamma R_2 + \ldots | S_0 = s, A_0 = a, \pi] = Q_{\pi}(s,a) = \sum_{s'} P_{s,a}^{s'}(R_{s,a}^{s'} + \gamma Q_{\pi}(s', a)) \tag{4.2}$$

A closely related concept is the state value function $V_{\pi}(S)$ which is the expected value of being in state s and following policy π, see appendix B.1.

Q-learning is concerned with finding the optimal action value function $Q^*(s,a)$ and its associated policy [4] π^* which is the solution to Equation 4.3. Once Q^* is known, the optimal policy can be derived on the fly by choosing the $a*$ which maximises $Q^*(a,s)$ when in state s.

$$Q^*(s,a) := \max_{\pi} Q_{\pi}(s,a) = E_P[r_{t+1} + \gamma \max_{a'} Q^*(s_{t+1}, a') | s_t = s, a_t = a] = \sum_{s'} P_{s,a}^{s'}(R_{s,a}^{s'} + \gamma \max_{a} Q_{\pi}(s', a)) \quad \forall s, a \tag{4.3}$$

[4] or policies since it may not be unique

Equation 4.3 is known as the Bellman optimality equation for Q^* (Sutton and Barto, 2018). The second term in the equality corresponds to the definition of the Q-function. The third term is an expansion of this definition over one period, and is a reformulation of Equation 4.2. The final term explicitly converts the expectation operator into its algebraic form with reference to the transition function $P^{s'}_{s,a}$.

The following update is used in one step classical classical Q-learning, and can be proven to converge in a finite action and state space to Q^* (Melo, 2001).

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha_t(r_t + \gamma \max_a Q_t(s_{t+1}, a) - Q_t(s_t, a_t)) \tag{4.4}$$

α controls the rate of update for each iteration of the Q-Value estimate and is reduced over time for convergence. The update uses a single observation of the reward r and subsequent state s' after choosing action a to update the state value Q(s,a). In other words the expectation in equation 4.3 is approximated with a one sample estimate. Miraculously, this works well in practice.

For large action and state spaces, $Q(s, a)$ is impractical to estimate directly. Instead it is typically approximated with a parameterised function $Q(s, a, \theta)$. The cost of tractability that a function approximator gives is that convergence is no longer guaranteed. Since neural networks are decent general purpose function approximators, they are a popular choice but are by no means the only one, see Chapter 8 in Sutton and Barto (2018). I will use a neural network and the following parameter update from Hasselt et al. (2016). Note that the s_t and a_t in practice are sampled batches of state and action analogous to normal updates in stochastic gradient descent.

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha(t)\Big(Y^Q_t - Q(s_t, a_t, \boldsymbol{\theta}_t)\Delta_{\theta_t} Q(s_t, a_t, \boldsymbol{\theta}_t)\Big) \tag{4.5}$$

Where $\alpha(t)$ is a declining update size and Y_t^Q is the known as the target function and defined as:

$$Y_t^Q := R_{t+1} + \gamma \max_a Q(s_{t+1}, a, \boldsymbol{\theta}_t) \tag{4.6}$$

The derivation behind this update is shown in Appendix B.2, page 245. At root it is motivated by a least squares error minimisation between the optimal state-action function Q^* and its parameterisation $Q(\cdot, \cdot, \boldsymbol{\theta}_t)$[5].

4.2.3 Dyna-Q

Dyna-Q is an augmented type of Reinforcement Learning whereby the RL learner, learns both through direct interaction with the environment and through simulation with an internal model of the environment, built from experience.

The Dyna-Q model offers flexibility over the choice of RL learning technique for the direct and simulated learning process and the type of supervised learning for the model fitting process. Introduced in Sutton (1990), it initially used tabular learning for learning and modelling. This was updated to linear function approximation in Sutton et al. (2008), and deep Q style learning in Peng et al. (2018). Deep Dyna Q means that in both the direct learning and simulated learning part of the process the agent learns through Deep Q learning using experience replay (Mnih et al., 2015). This means fitting a neural network $\tilde{Q} : S \times A :\rightarrow \mathbb{R}$ to estimate the value function Q^*. This is achieved through batch sampling of the experience replay memory and single step backups. I chose a family of neural networks known as Conditional Variational Autoencoders or CVAEs (Doersch, 2016). These are a generative deep learning technique which allow the user to fit a distribution to the process to

[5] An circular formulation since Q^* is unknown and the finding it is the greater objective of the exercise!

be modelled (instead of a point estimator). This is suitable for this application because the market dynamics are highly stochastic and the technique has the capacity to fit multimodal distributions (which the market is likely to produce). Applying this technique to Dyna-Q was introduced in Moerland et al. (2017). See Appendix B.6 for a quick introduction to CVAEs.

I use the deep Q variant but also a tabular approach for reference. I also test a hybrid approach where the model learnt by the agent is the empirical one, but the Q function remains a neural network.

1. Initiate Q network and Model network. Optionally initiate Reward, Done networks
2. Do until stopping criteria:
   (a) **Acting** Until episode is in terminal state:
      i. $s_0 \leftarrow$ current state, episode reward $r_{episode} = 0$
      ii. Choose action $a \leftarrow \epsilon$-greedy Q(s,a)
      iii. Execute action a, observe state s' reward r and done (terminal state) indicator $d \in \{0, 1\}$. $r_{episode} += r$, Store s, a, r, s', d
   (b) **Direct learning** Do deep Q learning:
      i. Sample batch from memory: $\mathbf{s}, \mathbf{a}, \mathbf{r}, \mathbf{s'}, \mathbf{d}$
      ii. Calculate target value: $\hat{Q} := \mathbf{r} + \gamma.\mathbf{d}.max_{\mathbf{a}'}Q(\mathbf{s'}, \mathbf{a'})$
      iii. Update Q network: Using stochastic gradient descent to minimise: $\mathcal{L}(Q(\mathbf{s}, \mathbf{a}), \hat{Q})$ for some appropriate loss function $\mathcal{L}$
   (c) **Update Model (unless tabular learning)**
      i. Supervised training of (Transition) Model (and Reward, Done models if using) using batches from memory.
   (d) **Planning through simulation** Repeat for N batches:
      i. Batch Sample **s** and **a** from memory.
      ii. Predict $\hat{\mathbf{s}'} \leftarrow model(\mathbf{s}, \mathbf{a})$
      iii. (Optional[a]) Predict $\hat{\mathbf{d}} \leftarrow Done(\hat{\mathbf{s}'})$ and $\hat{\mathbf{r}} \leftarrow Reward(\hat{\mathbf{s}'})$
      iv. Deep Q learning as in step 2b with batch: $(\mathbf{s}, \mathbf{a}, \hat{\mathbf{r}}, \hat{\mathbf{s}}', \hat{\mathbf{d}})$
   (e) Reset Environment. Reduce exploration and learning parameters.
   (f) Check stopping criteria

[a]The reward and done variables are deterministic functions of new state only. Giving the agent access to them simplifies the learning task.

Algorithm 1: Dyna Q algorithm

4.2.4 Model learning

I tried two formulations for the Agent's internal model:

1. **Stochastic model** The agent learns a stochastic neural network $f_\psi : S \times A \times Z \to S$, where $Z = \mathbb{R}^d$ is a latent space of user defined dimension. At prediction time, the latent space is sampled from at prediction time to inject stochasticity into predictions. This function is a direct approximation of the state transition $P(s'|s,a)$ for $s, s' \in S$ and $a \in A$.
2. **Tabular Learning** By recording every state encountered, the counts of actions taken therein and the counts of resulting states and rewards, the agent can build a sample estimator of the transition function. This is the simplest method and is only viable in a setup with a limited number of states. Simulation is limited to state, action, next state triples that have been experienced. Knowing an estimate of the transition function $P(s,a)$ does allow the possibility of full back ups as apposed to sample backups.

A motivation for the stochastic model provided by the CVAE can be seen in figure 4.2. In the situation where outcomes are multimodal, a deterministic model of the sort created by a vanilla neural network can cause trouble because they may end up predicting outcomes which never occur by averaging between common modes.

One problem with fitting a model using a neural network is that our state space is integer based. The model will make real value predictions for the next period but these predictions are unlikely to look like any state that the agent has encountered before. There is no guarantee therefore that the agent learns anything useful when simulating the environment with a model based on a neural network.

Integers and rational numbers are countably infinite and in our context finite. One approach might be to one hot encode every possible state but this would

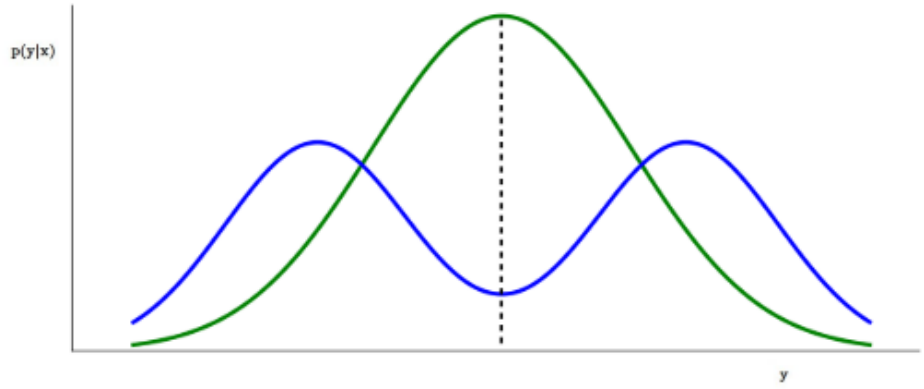


Figure 4.2: Non stochastic models trained with loss functions like MSE will misfit multimodal distributions by always predicting conditional mean - the dotted line. Figure from Moerland et al. (2017).

lead to a neural network with very large layers which may be very difficult to train. Another approach is to apply a rounding function to the output of the neural network. This is fine on the forward pass but is not differentiable and therefore ineligible for gradient descent on the backward pass. I found no satisfactory solution to this problem when using neural networks to create a transition model for the agent.

The traditional tabular approach to constructing a model from experience constructs transition probability tables from experienced (state, action, reward, subsequent state) transitions. Its feasibility is limited by the size of the state action space and will only predict transitions that are in memory. However it constructs an empirical distribution estimator for every transition which is robust to multi-modality and its maintenance has little overhead.

4.3 Method

The experiment consists of a learning task for one reinforcement learning agent, trading in a market populated by a variety of other zero intelligence traders.

At the beginning of each episode, the agent starts with an inventory of one. Their task is to sell the inventory at the best possible profit at which point the

episode ends. They are not allowed to accumulate more inventory and if they do the episode also ends. The agent is allowed to sell their inventory at best bid, post limit orders on the bid side, do nothing and cancel any outstanding (bids) they have on exchange. There is also a stop loss which ends the episode if the unrealised gain/loss of the agent becomes too large. Each episode can continue for at most 100 periods. These rules define the termination function in equation 4.10 on page 80.

At the end of every episode, the agent's holdings are liquidated at best bid, and any outstanding orders are cancelled. The simulation environment is then progressed without the RL trader for 100 periods after which I assume their effect on the market has been forgotten. This procedure is detailed in Box 3 on page 83.

The objective of the experiment is to see whether the agent learns a profitable strategy trading which relies on the posting of bids (which it does not want to execute). The return distribution of this strategy is then compared with one where the agent does not have the ability to submit orders on the bid side, and various user defined benchmark strategies such spoofing strategy and a passive strategy where the trader buys and holds for the duration of the experiment.

4.3.1 Market Environment

I consider a market for a single asset populated by four different types of agents. The setup proceeds in a similar way to Wang and Wellman (2017). At the beginning of each market session, an underlying fundamental price sequence is generated of the form:

$$r_t = max\{0, \kappa\bar{r} + (1-\kappa)r_{t-1} + u_t\} \quad \text{for } u_t \sim N(0, \sigma_s^2),\ \ \kappa \in (0,1) \tag{4.7}$$

For our experiments I chose values of $\kappa = 0.0002$, noise variance $\sigma_n^2 = 1$ and

mean price $\bar{r} = 100$.

A 'tiled' version of this price sequence was then used where each price point in the sequence is repeated for $b = 10$ periods. This is described in Equation 4.8. This feature was added to develop the idea of information flowing more slowly to informed market participants than the potential acting frequency of the traders within it.

$$\tilde{r}_t = r_{t \bmod b} \quad \text{for } t \in [0, \ldots, t_{max}] \tag{4.8}$$

On spawning, all agents are assigned a preference function over holding units of the asset long or short which exhibits diminishing marginal utility. This is a agent specific vector of $\Theta := (\theta^{-q_{max}}, \ldots, \theta^{q_{max}})$ where q_{max} is the maximum quantity of asset that an agent can hold. θ^{q+1} can be interpreted as the marginal utility in buying another unit whilst already holding q units, see also discussion of Wang and Wellman (2017) in Section 2.1.3 on page 45. For our experiments all traders had inventory limits of $q_{min} = -5$ and $q_{max} = 5$.

4.3.2 Trading Agent types

The Market environment was populated by 10 instances each of the following four types of trading agent:

1. Wang Wellman Zero Intelligence Traders (WWZI) as found in Wang and Wellman (2017) and described in section 3.7.1.1, page 58. These traders receive a private noisy signal about the fundamental, update their prior beliefs as Bayesians and place orders based on their prediction of the future (10 periods) fundamental price.
2. Heuristic Belief Traders (HBL) as found in Wang and Wellman (2017), described in section 3.7.1.2, page 58. As with WWZI traders, this class also maintain a probability function of an order executing as a function

of price. Order prices are chosen to maximise the expected surplus over the estimated fundamental price.

3. Noise Traders (NOI), described in section 3.7.1.3 on page 60. These traders place orders according to the recent quantity weighted average order submission, adjusting for any cancellations.
4. Imbalance Traders (IMB) as described in section 3.7.1.4 on page 60. These traders place orders based using a submission price derived as a linear function of Order Flow Imbalance as defined in Cont et al. (2013). The memory of these traders is drawn at random on spawning from uniform distribution $U[5, 15]$.

These agents all transact in single units. This is a result of their simple trading strategies not a market restriction; indeed the RL agent is free to place multi quantity orders. Time is divided into discrete units; for our experiments a market session consisted of 6000 time units. Price is restricted to integer values. The agents do not incur transaction costs.

4.3.3 Reinforcement Learning Environment

In this section I will describe the fundamental parts of the Reinforcement Learning environment: State space, Action space, Termination Reward functions.

4.3.3.1 Action space

The agent has five possible actions at time t:

$$a_t^i = \begin{cases} i = 0 & \text{Do nothing} \\ i = 1 & \text{Cancel bids} \\ i = 2 & \text{Add bid at best bid, quantity 1} \\ i = 3 & \text{Add bid at best bid, quantity 5} \\ i = 4 & \text{Sell inventory at best bid} \end{cases} \tag{4.9}$$

Action 4 - selling inventory will always end the training episode since inventory will become zero.

4.3.3.2 State Space

I chose the state space to be the simplest space possible to allow the emergence of price manipulation. Variables were scaled to lie close to the origin for the benefit of the various neural networks which received the states as inputs. A description of the states and their scaling is given in table 4.1.

Name	**Description**	**Scaling**
$Dist_t$	Negative cumulative change in best bid price from entry	0.1 mult clipped [-1,1]
$Invent_t$	Inventory	Effectively 0,1,2
$Orders_t$	Number of separate orders in market	Binarised >0
$BASpr_t$	Current spread between best bid and ask	0.1 mult clipped [0,1]
ΔBid_t	Period on period change of best bid	
ΔAsk_t	Period on period change of best ask	
PIL_t	Position in Lob: Distance in quantity order from trader's closest order to front of best bid queue. Takes value -1 when no orders in market.	0.5 mult clipped [-1,2]
$Imbal_t$	Order imbalance of orderbook	0.05 clipped [-1,1]
$Time_t$	1 final period, 0.5 penultimate period, 0 else.	0,0.5,1

Table 4.1: State space used with a description of the scaling and censoring applied.

4.3.4 Reward and Termination function

The training episode ends when the terminal function equals one:

$$d(\underline{s}, Cutoff, Ub, Lb) = \begin{cases} 1 & \text{if } invent_t <> 0 \\ 1 & \text{if } time_t => Cutoff \\ 1 & \text{if } Dist_t => Ub \text{ or } Dist_t < Lb \\ 0 & \text{else} \end{cases} \tag{4.10}$$

A training episode ends when the trader's inventory is zero (implying a sell trade has happened) or inventory is greater than one (implying a buy trade has happened). Additionally an episode ends when the unrealised profit of their position is less than some lowerbound $Lb = -2$ or greater than an upper bound $Ub = 10$. Finally, a training episode if none of the other events have happened after a $Cutoff = 100$ periods.

The agent has the following reward function:

$$r(\underline{s}_t) = \lambda \Delta bid_t + \begin{cases} (1-\lambda)Dist_t - \lambda \Delta bid_t & \text{if } Invent_t = 0 \\ -(Invent_t - 1) * BASpr_t + (1-\lambda)Dist_t & \text{if } Invent_t > 1 \\ -(1-\lambda)Dist_t & \text{if otherwise terminal: d(s,Cutoff,Ub,Lb)=1} \\ \delta_o \mathbb{1}(Orders_t > 0) & \text{else} \end{cases}$$

The reward function is a period wise approximation of profit. If the agent has positive inventory, the agent receives the best bid change every period as a reward weighted by a factor $\lambda \in [0,1]$. On completion of an episode, the agent receives $(1-\lambda)Dist_t$ where $Dist_t$ is the total change in best bid over the lifetime of the trade. The reward function is therefore equal to cash profit less the initial bid ask spread. The coefficient λ weights unrealised and realised profit. It is designed to approximately telescope out at the end of every episode. If the episode ends through the agent accidentally buying more stock (exceeding inventory limit), they are penalised by the current cost of

liquidating their excess position; the bid ask spread multiplied by their excess inventory.

Aside from the period wise approximation of realised profit, the agent also receives a small reward δ_0 for having orders in the market. This and the piece-wise nature of the reward function allows better training than simply rewarding or penalising the agent at the end of the episode.

Since some termination conditions involve position liquidation outside the experiment episode (with uncertain outcome), there is a degree of approximation to the reward function which is difficult to escape. Similarly the acquisition of inventory is also done within the simulation environment but outside the experiment episode in period t=-1. This means that whilst the inventory is always bought when bid ask spread is equal to one, this may change at period $t = 0$. To be consistent across all values of λ, the agent's initial reward includes $r(\underline{s}_0) = \lambda \Delta bid_0$. Consequently it may receive credit or a penalty before actually doing anything.

4.3.4.1 Exploration policy

Reinforcement learning requires learning agents to have a policy which includes an element of exploration of new state-action pairs over time. Indeed the proof of convergence of temporal difference learning of which Q-learning is a variant, requires that all states are visited infinitely often. I chose a variant of the typical $\epsilon - greedy$ policy whereby our agent would choose the least chosen action in that state with probability ϵ. Over time, ϵ was decayed from an initial value of 0.8 to a base value of 0.01 with the effect that the RL Agent would increasingly choose to 'exploit' their existing optimal policy as defined in the state action function $Q_t(s, a)$.

In Experiment 4, I followed an optimism in uncertainty strategy based on the UCB algorithm (see appendix B.7 for an explanation of this). I found that

this improved exploration and lowered the chance of premature convergence. UCB is increasingly being used instead of $\epsilon - greedy$ exploration and is shown to achieve good sample efficiency by Jin et al. (2018).

4.3.5 Training routine and sub-routines

In this section there is a description of the key component procedures envoked during the training process. *RL Agent: Choose Action* describes how the RL agent chooses actions when learning. *RL Environment: Reset process* describes the process which occurs to ready the Market Environment for the the RL Agent to begin interacting with it. *RL Environment: Step process* describes what happens once the RL Agent chooses an action. *Market Environment: Simulate One Period* describes the process through which the other traders in the simulation respond after the RL agent has acted. Finally *RL Environment: Training Process* describes the end to end training process which unites these subroutines.

Input variables:
- ϵ_t - probability of not following greedy strategy
- Estimate of state action function $Q(\cdot, \cdot, \theta_t)$

1. Draw h from uniform distribution $U[0, 1]$
2. if $h > \epsilon_t$:
   - choose $a_t = \max_a Q_t(s, a, \theta_t)$
3. Else:
   - Choose a_t such that $a_t = \min_a N(s, a)$ where $N(s, a)$ is the number of times a has been chosen in state s

Algorithm 2: RL agent action choosing

4.3.6 Optimal policy determination

I found it necessary to test the performance of the optimal strategies derived from $Q_t(\cdot, \cdot, \theta_t)$ separately rather than rely on measures of reward derived during online training. The reasons for this are threefold. Firstly, when ϵ was greater than zero, the RL trader would conduct policy exploration online.

Input variables:
- Market Environment time $t \in \{0, \dots, T = 6000\}$
- Market Environment $timeRemaining_t := T - t$
- Trader inventory $invent_t$

1. Do until $invent_t = 0$: Liquidate RL trader inventory and cancel outstanding orders.
   (a) RL Agent: Submit orders of appropriate quantity at best bid or ask to zero inventory
   (b) RL Agent: Cancel all outstanding orders
   (c) *Market Environment: Simulate One Period*
   (d) t=t+1
2. If *timeRemaining* in $session < 2 * sessionLimit$: Generate New Market environment:
   - Make new fundamental price sequence $\{p_t\}_{t=0}^{6000}$,
   - Spawn new traders $\{Z_i\}_{i=0}^{i=40}$ with new trader specific parameters: inventory preferences Θ_i etc.
3. Until LOB of adequate depth and bid ask spread =1:
   - *Market Environment: Simulate One Period*
   - $t = t + 1$
4. RL agent buys one unit at best bid.

Output variables:
- Initial state s_0
- Initial reward r_0

Algorithm 3: RL environment reset process

Input variables:
- RL Agent action $a_t \in A$

Begin:
1. RL trader does action a_t: If this is not $a_t = 0$: 'do nothing':
   (a) Submit order or cancellation to exchange
   (b) Order is processed: if it crosses best bid or ask, trade is executed and counterparties informed.
2. *Market Environment: Simulate one period*
3. New state of RL Environment s_{t+1} is generated.
4. New reward of RL environment r_{t+1} is calculated.
5. Check if s_{t+1} is terminal condition.

Output variables:
- New state s_{t+1}
- New reward r_{t+1}
- Terminal indicator $d_{t+1} \in \{True, False\}$

Algorithm 4: RL environment step process

Set of traders $\{Z_i\}_{i=0}^{i=40}$, period in Market Environment time $t \in \{0,\ldots,6000\}$

1. Traders receive new state of LOB from exchange - update internal variables accordingly.
2. Choose trader z_t at random according to distribution $P(Z)$.
3. Inform trader of private signal if trader is of type WWZI, HBL.
4. Trader submits order(s) to exchange.
5. Orders are executed by exchange, traders informed of trades.

Algorithm 5: Market environment: Simulate one period

1. for episode $i = 1,\ldots,I = 10,000$:
   (a) Initiate episode variables: $Reward_i = 0$, $t = 0$
   (b) $s_0, r_{-1}, d_0 =$ *RL Environment: Reset*
   (c) While not d:
      i. a_t^i=*RL Agent: Choose action*(s_t)
      ii. s_{t+1},r_t,d=*RL Environment: Step*(a_t^i)
      iii. Store $(s_t, a_t, r_t, d, s_{t+1})$ in Memory
      iv. $Reward_i$=$Reward_i + r_t$, $s_t = s_{t+1}$, $t = t+1$
   (d) Store $(i, t, Reward_i)$
   (e) **Direct Learning**: Do Q learning on batch sampled from Memory.
   (f) If Model and DynaQ: **Model based Learning**:
      i. **Supervised model training**: $argmin_F\, d(F(X), Y)$ for:
         - **Transition Model**: $x \subset \{s_t, a_t\}$, $y \subset \{s_{t+1}\}$
         - *Optional*[a] **Reward Model**: $x \subset \{s_{t+1}\}$, $y \subset \{r_t\}$
         - *Optional* **Termination Model**: $x \subset \{s_{t+1}\}$, $y \subset \{d_t\}$
   (g) If Model: **Model based learning**:
      i. Repeat for batch size:
         - Sample s_0, *RL Agent: Choose Action* a_0
         - get $\hat{r}$,$\hat{s}_+$,$\hat{d}$ from model $F(s_0, a_0)$
         - Do Q learning update

[a] Alternatively the Reward and Termination functions are given to RL agent as functions of s_{t+1} which is estimated through $\hat{s}_{t+1} = F(s_t)$

Algorithm 6: RL environment: Training process

Since choosing action 4 meant that the trading episode would cease immediately with an inventory liquidation, this affected the performance of any strategy under consideration. Secondly and most importantly, the environment was highly stochastic so any assessment of a strategy would be variable unless assessed over a number of trading episodes. Finally, the performance of the training reward was not monotonic (except in pure tabular learning of the Q function). Optimal strategies were no more likely to appear towards the end of training than the beginning. Alternatively put, the Q-function did not converge in general.

Unfortunately, strategy testing was time consuming since it required interacting with the trading simulator. The testing of every iteration of the Q_t was therefore impractical. As a balance, strategies would be tested only if their recent mean reward was close to the best of the experiments. If this condition was met, the strategy was tested with zero chance of exploration over 20 trading episodes. If the mean reward of this test was higher than the recorded best strategy's, a further test of 40 trading episodes was conducted. If after this the mean reward (over the combined 60 episodes) was better than the best, this new strategy would become the best. At the end of the experiment (10,000 trading episodes), the performance of the best strategy was further assessed over 5,000 trading episodes.

4.3.7 Benchmarking and method of assessment

I measured our RL derived trading algorithms against three strategies:

1. **Passive**: A totally passive trading strategy that sells when profit is below a lower bound (-2) and profit is at or above a profit target $P \in (0, UB)$. I tested a number of values for P. $P = 10$ was too rare an occurrence, $P = 2$ was the most profitable on a probability weighted basis.
2. **Submit order**: A strategy where the agent would just submit bid orders of quantity 1 or 5 each period. This is included to assess whether there

is some skill in RL derived optimal strategies.

3. **Simple spoofing**: Agent will submit bid orders when their position in LOB (PIL_t) is above 0. In other words, when there is a lower chance of execution in the next period. When profit has reached a profit target 2 or declined below -2 as before, agent sells inventory at best bid and cancels outstanding orders.

I also trained the RL agent with a constrained action space where they could only wait or sell their inventory at best bid. This *Constrained Action Space* precludes the agent from learning any market abusive practices. It is included to test the ability of the learning algorithm to identify sensible strategies using the provided state space but without the possibility of manipulating the LOB.

4.3.8 Neural network architecture, loss functions and training

4.3.8.1 Q function Approximation

In all but one experiment, I approximated the Q-function $Q^{\pi^*}(S, A) \rightarrow \mathbb{R}^{|A|}$ with a neural network comprising two hidden layers of 16 fully connected neurons, each separated by a relu non-linearity function (Goodfellow et al., 2016). Training was done with the Adam variant of stochastic gradient descent (Kingma and Ba, 2014). A batch size of 64 was chosen throughout for updates. Batches were made possible by using 'experience replay' as used by Mnih et al. (2015). Experience replay refers to the storage and sampling of previously experienced transitions. Sampling and updating from this de-correlates updates (which would occur if updating from a full episode of experience). This aids with stability and data efficiency (since experience is reused over training)

In all the Deep-Q experiments I followed the double-Q method of training, first introduced by Van Hasselt (2010). This is particularly important in this application because the environment is highly stochastic. The author of this paper finds that in this case, the traditional Q update over-estimates action

values. In practice this means that two networks value action networks are maintained - a target network and an evaluation network. See appendix B.3 for an explanation of double Q learning.

4.3.8.2 CVAE design

A CVAE can be thought of as an autoencoder with encoder and decoder networks. I used a simple encoder network with input of size $2|S|+2$ leading to a layer of 16 fully connected hidden units, a Relu layer for non-linearity and two outputs for mean and variance in a latent space of two dimensions. The decoder took as input the mean and variance parameters of latent dimension 2 plus the state space $|S|$ plus and a one hot encoding of the action space (corresponding to the conditional space (s,a)). This was connected to a fully connected layer of 16 hidden units, a Relu layer before an output layer of size $|S|$. This output formed the network's prediction of the next state s'. Two separate, fully connected layers took this prediction of state to a prediction of reward r and terminal indicator d.

I chose $\alpha = 0.1$, and $\beta = 1$ to weight the reconstruction and KL errors in the objective function to train the CVAE (equation 25). I chose reconstruction error to be Mean Squared Error aka l_2 loss.

4.4 Results

Table 4.2 describes the design of the four experiment configurations I tested. Each experiment has a version where there is a full action space and a constricted action space consisting only of a wait and lift bid (sell inventory and finish). This is to compare the profitability of strategies where spoofing is possible and where it is not. The constricted action space precludes spoofing because the agent has no method to send signals to the other market participants via the order book. All experiments bar one used Q-learning with sample updates to the Q-function as described in Appendix B.2. Experiment

4 uses a 'full backup' as described in Chapter 9 of Sutton and Barto (2018) where the value of each next possible state, weighted by the probability of transitioning into that state, is used to update the value of being in state s and taking action a. Full updates like this are not compatible with neural network approximation of the Q-function and are only possible in this experiment because this is an application of tabular learning wherein the agent maintains a frequency table of each observed state transition.

Experiment #	Q Function	Backup	Model	Reward Model	Terminal Model	Action Space
Exp 0	Neural network	Sample	None	N/A	N/A	Full
Exp 0_0	Neural network	Sample	None	N/A	N/A	Do nothing, Cross bid
Exp 1	Neural network	Sample	CVAE transition	Trained	Trained	Full
Exp 1_0	Neural network	Sample	CVAE transition	Trained	Trained	Do nothing, Cross bid
Exp 2	Neural network	Sample	CVAE transition	Given	Given	Full
Exp 2_0	Neural network	Sample	CVAE transition	Given	Given	Do nothing, Cross bid
Exp 3	Neural network	Sample	Tabular transition	Given	Given	Full
Exp 3_0	Neural network	Sample	Tabular transition	Given	Given	Do nothing, Cross bid
Exp 4	Tabular	Full	Tabular transition	N/A	N/A	Full
Exp 4_0	Tabular	Full	Tabular transition	N/A	N/A	Do nothing, Cross bid

Table 4.2: Summary of experiment design. 'Full' action space refers to 5 dimensional action space. Each configuration is tested against a limited action space where spoofing is impossible; these experiments have a '0' subscript as an identifier. 'Sample' backup refers to the traditional Q-learning update. 'Full' refers to the Empirical Q-value iteration update (see Appendix B.5

Tables 4.3 and 4.4 show the profitability of the best strategies found in each experiment during training. This is shown graphically in figure 4.3. Profit refers to the accounting profit of the trade and adjusted reward is the reward attained by the agent during the episode. There is a difference of approximately one between each measurement as there was an initial spread of 1 when the trader's initial inventory was first acquired. Reward is designed to be a measurement of best bid improvement. A further difference between the two measures also occurs if an agent finishes an episode with non zero inventory. In this case liquidation is carried out automatically, if inventory is greater than 1, this could be costlier for the trader than the reward function gives credit for. To allow comparison between restricted and unrestricted action space experiments, the cumulative reward of the later is adjusted by deducting the bid placing bonus of 1/250 multiplied by the number of periods that the strategy continued for. An alternative would have been to not include the bid bonus in the reward

function but I found that it greatly aided exploration. This was evident in the training of the experiments with the constricted action space with no bid bonus. This made these experiments more likely converge to the strategy of selling in period one of the episode. No different experience is gathered and the agent learns nothing.

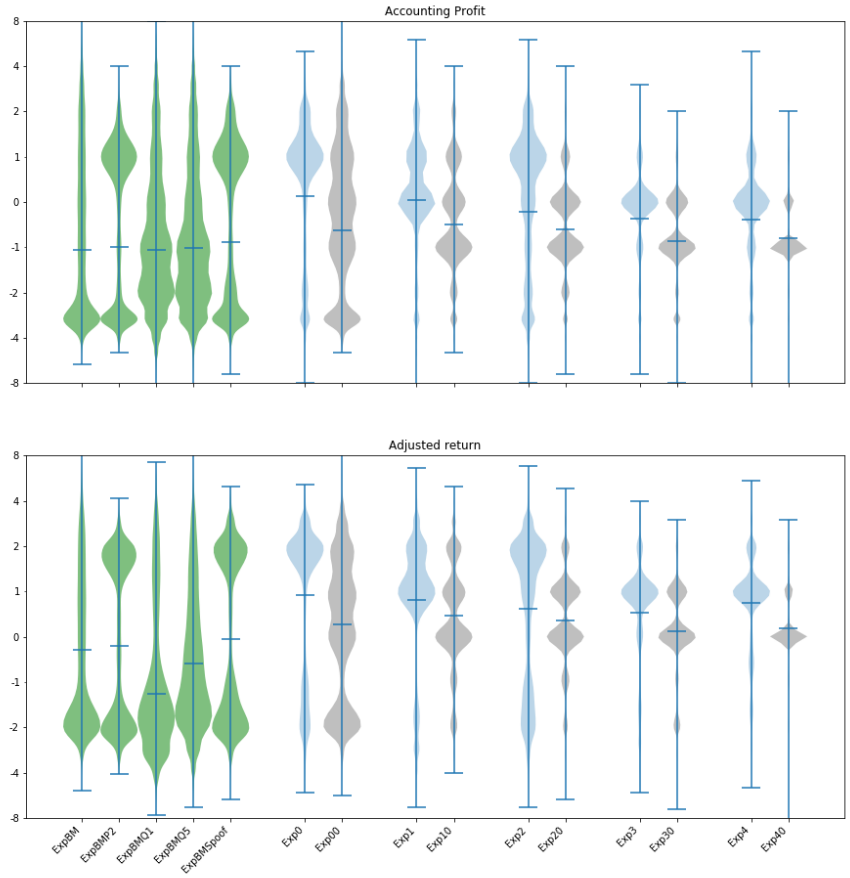


Figure 4.3: Profitability and reward distribution of best found strategies. Note log scale on y-axis. Benchmark strategies in green, full action space in blue, constrained action space in grey. All experiments show an improvement over benchmark strategies and constrained-space partner experiment.

I find that in all experimental settings, the trader with the larger action space is more profitable. An optimal strategy must always be at least as good as another strategy found with a limited action space. Opposing this is is also

countervailing force of a larger action space making the search for optimal strategies exponentially harder. Nevertheless, this indicates that the trader has found that the posting of bid orders is beneficial to their final reward. Also note that the best strategy found in the limited action space (Experiment 1.0) is worse than the worst strategy in the full action space (Experiment 4 for profit, Experiment 3 for adjusted reward) for both profit and adjusted reward.

_	BMQ0	BMP2	BMQ1	BMQ5	BMSpf	Exp0	Exp00	Exp1	Exp10	Exp2	Exp20	Exp3	Exp30	Exp4	Exp40
count	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000
mean	-1.07	-0.99	-1.06	-1.02	-0.9	0.12	-0.62	0.05	-0.5	-0.21	-0.62	-0.37	-0.87	-0.39	-0.79
std	2.50	2.00	2.11	2.15	2.06	1.85	1.96	1.53	1.27	1.85	1.05	1.21	0.96	1.36	0.61
min	-6.00	-5.00	-9.00	-9.00	-7.00	-8.00	-5.00	-17.00	-5.00	-8.00	-7.00	-7.00	-8.00	-12.00	-13.00
25%	-3.00	-3.00	-2.00	-2.00	-3.00	-1.00	-3.00	0.00	-1.00	-2.00	-1.00	0.00	-1.00	-1.00	-1.00
50%	-2.00	-1.00	-1.00	-1.00	-1.00	1.00	-1.00	0.00	-1.00	1.00	-1.00	0.00	-1.00	0.00	-1.00
75%	1.00	1.00	0.00	0.00	1.00	1.00	1.00	1.00	0.00	1.00	0.00	0.00	0.00	0.00	-1.00
max	9.00	4.00	8.00	14.00	4.00	5.00	9.00	6.00	4.00	6.00	4.00	3.00	2.00	5.00	2.00

Table 4.3: Accounting profits of the RL traders. Each experimented was repeated with a limited action space where action space is restricted such that manipulation is not possible (highlighted grey). Performance is adversely affected.

_	BMQ0	BMP2	BMQ1	BMQ5	BMSpf	Exp0	Exp00	Exp1	Exp10	Exp2	Exp20	Exp3	Exp30	Exp4	Exp40
count	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000
mean	-0.29	-0.02	-1.25	-0.60	-0.06	0.93	0.28	0.82	0.46	0.61	0.37	0.53	0.11	0.74	0.19
std	2.18	1.95	2.13	1.99	2.06	1.92	1.81	1.76	1.23	1.90	1.02	1.26	0.93	1.16	0.58
min	-5.25	-3.97	-7.69	-6.77	-6.01	-5.46	-5.71	-6.76	-4.00	-6.76	-6.00	-5.39	-7.00	-5.00	-11.75
25%	-1.97	-1.96	-2.76	-1.78	-2.01	-1.07	-1.87	0.97	0.00	-1.19	0.00	0.24	0.00	0.93	0.00
50%	-1.36	-0.01	-1.77	-1.10	-1.07	1.92	0.13	0.99	0.00	1.48	0.00	0.99	0.00	0.99	0.00
75%	1.31	1.95	-0.42	0.57	1.93	1.98	1.81	1.98	1.00	1.96	1.00	0.99	1.00	0.99	0.00
max	8.80	4.21	7.27	14.83	4.97	5.11	9.03	6.62	4.97	6.87	4.81	3.98	3.01	5.45	2.98

Table 4.4: Adjusted rewards of RL traders: Adjustment consists of deducting 1/250 for every period in a RL trader episode - to account for reward mismatch vs restricted action RL traders who cannot get the 1/250 bid submission bonus.

Tables 4.3 and 4.4 show that there is no obvious advantage to learning with the Dyna-Q algorithm over plain deep Q learning, although there is some benefit in the constrained space.

Table 4.5 shows the duration distributions of the final strategies. The maximum allowable trading duration was 100 periods. Longer strategies are penalised by the trader's inter-temporal discount rate. Insight into the reason behind the termination of trading episodes can be seen in table 4.6. In all experiments, the most common episode ending is for the agent to voluntarily finish by liquidating their inventory at best bid. This could indicate that

Rl traders have learnt how to exploit profitable situations or avoid bad ones. This is supported by Experiment 1 which finds a strategy that never ends from the risk limit being breached. Conversely this might also be indicative of premature exploitation in favour of future exploration.

_	BMQ0	BMP2	BMQ1	BMQ5	BMSpf	Exp0	Exp00	Exp1	Exp10	Exp2	Exp20	Exp3	Exp30	Exp4	Exp40
count	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000	5000
mean	70.42	45.26	53.51	55.22	40.30	30.94	45.17	21.44	20.03	30.66	14.98	19.76	11.83	17.96	4.98
std	34.95	30.54	32.94	32.37	27.81	21.77	29.88	14.57	18.29	21.29	15.43	20.81	16.08	14.93	7.75
min	1	1	2	2	1	2	1	2	1	1	1	1	1	1	1
25%	35	20	25	27	18	15	21	11	7	15	4	7	1	8	1
50%	100	37	44	48	33	25	39	18	15	27	10	12	5	14	1
75%	100	67	95	95	56	39	66	28	28	42	21	24	17	23	6
max	100	100	100	100	100	100	100	100	100	100	100	100	100	100	97

Table 4.5: Duration of trading episode (limit 100 periods)

The proportion of times that an episode ends with inventory greater than 1 is in indicator of how successful a spoofing strategy is since a spoofer should never *intend* their bid orders to be executed. This level is at about 20% for the most profitable Experiments 1-3.

_	BMQ0	BMP2	BMQ1	BMQ5	BMSpf	Exp0	Exp00	Exp1	Exp10	Exp2	Exp20	Exp3	Exp30	Exp4	Exp40
-distance <lb	0.49	0.44	NaN	NaN	0.36	0.1	0.29	NaN	0.08	0.09	0.06	0.02	0.1	0	0.01
-distance >ub	0	NaN	NaN	0	NaN	NaN	0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
inventory 0.0=0	NaN	0.43	NaN	NaN	0.43	0.68	0.66	0.81	0.92	0.67	0.94	0.85	0.9	0.81	0.99
inventory 2.0>1	NaN	NaN	0.76	0.76	0.14	0.19	NaN	0.19	NaN	0.22	NaN	0.11	NaN	0.18	NaN
time up 1.0	0.51	0.13	0.24	0.24	0.07	0.03	0.04	0.00	NaN	0.02	0.00	0.02	0.00	0.00	NaN

Table 4.6: Proportions of how strategies end episodes. Inventory=0 row highlighted in bold corresponds to strategy choosing to end episode by lifting best bid and cancelling open orders

Strategies associated with spoofing should be those strategies which choose actions corresponding to the placement of bids. The proportion of actions chosen by strategy is shown in table 4.7. The submission of bids of quantity 5 is more aggressive than bids with quantity 1 since it will lead to a higher order book imbalance. The two best strategies found in Experiments 0 and 1 both eschew bid placement of quantity 1 for quantity 5. All strategies place more quantity 5 orders with exception of Exp' 3. Cancellations are also a feature of spoofing strategies. In all cases, the found strategies feature order cancellations. A complicating factor with cancellations for the trader is the negative effect that it has order book imbalance (and subsequent price movements). The benefit of cancelling all orders (avoid execution) is counteracted by the likely negative

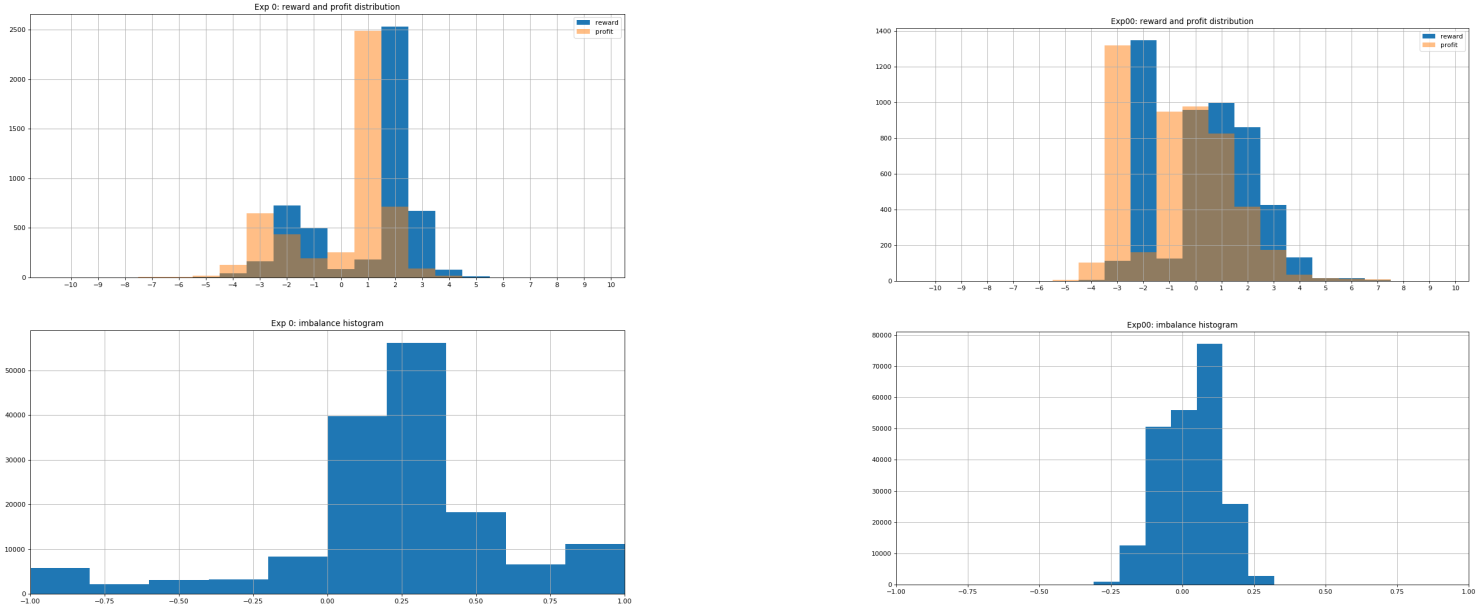
price impact it will have in proceeding periods.

_	BMQ0	BMP2	BMQ1	BMQ5	BMSpf	Exp0	Exp00	Exp1	Exp10	Exp2	Exp20	Exp3	Exp30	Exp4	Exp40
Do nothng	1.00	0.99	NaN	NaN	0.19	0.61	0.99	0.70	0.95	0.58	0.94	0.06	0.92	0.34	0.80
Cancel bids	NaN	NaN	NaN	NaN	0.02	0.02	NaN	0.01	NaN	0.03	NaN	0.01	NaN	0.02	NaN
Submit best bid, q=1	NaN	NaN	1.00	NaN	NaN	NaN	NaN	NaN	NaN	0.11	NaN	0.84	NaN	0.24	NaN
Submit best bid q=5	NaN	NaN	NaN	1.00	0.78	0.35	NaN	0.25	NaN	0.25	NaN	0.03	NaN	0.32	NaN
Finish: Lift best bid	NaN	0.01	NaN	NaN	0.01	0.02	0.01	0.04	0.05	0.03	0.06	0.06	0.08	0.08	0.20

Table 4.7: Mix of actions chosen in strategy.

The distribution of returns shown in left panel figure 4.4 for best policy found in Experiment 0 is multimodal; the left mode roughly corresponds to the outcome where inventory is accidentally acquired (19% of the time according to table 4.6 on page 91) and when the total negative move in best bid is more than 3 (10% of the time). The distribution of returns for the analogous experiment in the constrained action space is shown in the right panel of the same figure. Whilst also multi modal, the distance between the two modes is smaller; in this setting the agent is unable to influence the market by increasing order book imbalance. The difference in distributions of imbalance between the two experiments is pronounced.

The multi-modality of the return distribution is not present in all experiments. This can be seen in Appendix C.1 from page 260. It is not present for the least profitable experiments 3-4. Whilst I can explain why the return distribution might look multimodal, it is unclear where it is a necessary feature of a profitable spoofing strategy.



(a) Exp 0

(b) Exp 00

Figure 4.4: Upper charts shows distribution of returns associated with the best policy found in Experiments 0 and 00. Lower chart shows distribution of the order book imbalance metric during trading. The value shown is scaled by a factor of 0.05 and clipped to be in the range in [-1,1] hence the peaks at either end. Observe narrow distribution of imbalance in Exp00 Figure 4.4b where the trader has no ability to affect it.

93

4.4.1 Policy Interpretation through Tree fitting

For each optimal strategy found, I used the test data to fit a tree classifier (using CART algorithm and Gini coefficient as impurity measure) taking s as input and outputting a. The maximum depth of the tree was set to 4 and the minimum sample proportion for a leaf was set to 1.5%. The depth was limited to 4 to avoid over-fitting. Minimum sample proportion was set with reference to Table 4.7. In practice, the maximum tree depth was attained for most leaves. The data was split at random between training and test data with a 4:1 split. Because the actions were severely imbalanced[6], the action classes were reweighted to equality by weighting data points (not through over or under-sampling). The accuracy statistics reported below reflect this.

The obtained training and test fits (accuracy) are shown in table 4.8. The proximity of test errors to train errors supports the assertion that the trees were not over-fitted at this depth. Fit is adequate with the exception of Experiment 4. It is higher for the experiments with reduced action space which is expected. Lifting the depth restriction did not improve fit very much and certainly did not balance the decrease in interpretability.

_	**Exp0**	**Exp00**	**Exp1**	**Exp10**	**Exp2**	**Exp20**	**Exp3**	**Exp30**	**Exp4**	**Exp40**
train_score	0.90	0.91	0.75	0.94	0.70	0.97	0.83	0.84	0.49	0.96
test_score	0.90	0.90	0.74	0.94	0.70	0.96	0.82	0.83	0.48	0.96

Table 4.8: The found optimum strategies were approximated by a tree classifier, this table shows the accuracy scores of the tree. Action classes were given importance weights to be equal to take account of imbalanced action choices.

The normalised Gini Importance for each feature is shown in Figure 4.5. This is a statistic which measures the relative importance of each feature in the decision tree. I expected the most important features associated with spoofing to be *imbalance* and *position_in_lob*. *imbalance* because this is the feature that the RL trader can impact directly. It is the most important feature for Exp0, Exp1 and Exp4, it is second for Exp2 and third in Exp3. *position_in_lob*

[6] Crossing the bid to liquidate inventory could only happen once for example.

because it relates to the chance of the agent accidentally buying more inventory. It is important for experiments 0, 2 and 3 but not hugely for 1 and 4. I expected *distance* to be an important feature since it determines the returns to liquidating inventory and ending the trading episode. It is a surprise then that it does not feature highly in the full action space strategies except Exp 4. However it is important for the restricted action space strategies suggesting a better mastery of exploitation. *bid_ask_spread* and *ask_change* are selected as important features for nearly all strategies; this might indicate that it has some predictive power. *bid_change* can be inferred by these two.

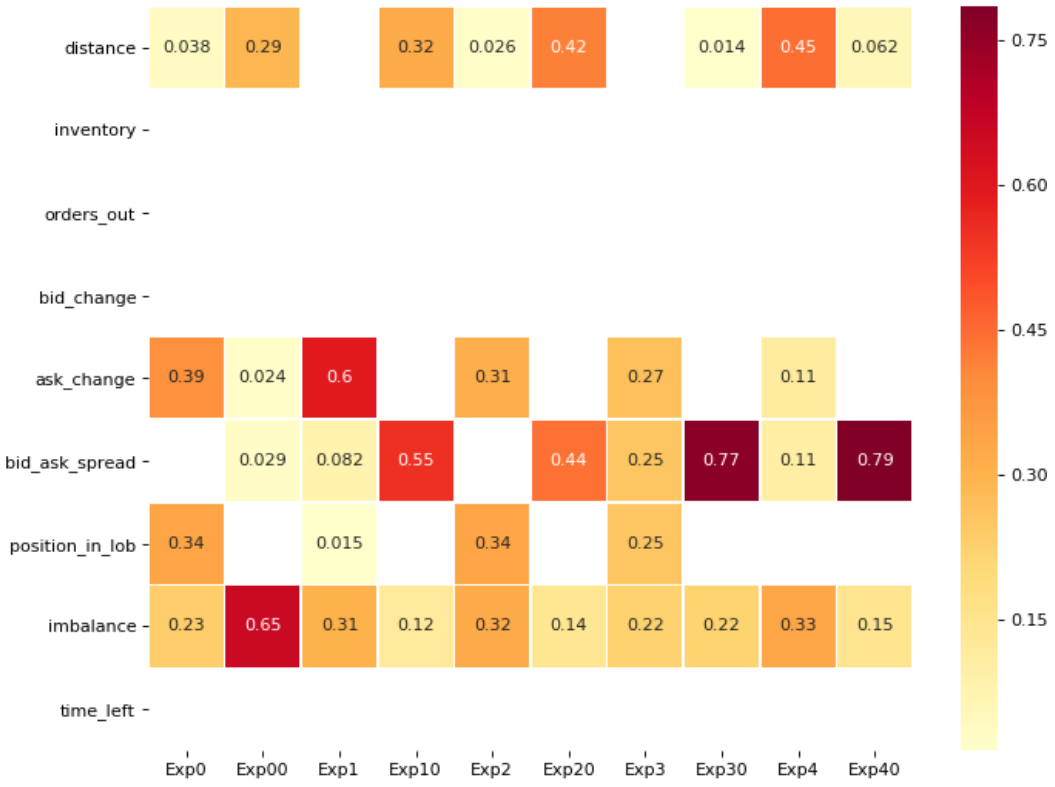


Figure 4.5: Heat map showing normalised Gini Importance of features of decision trees approximating found optimum strategies. Strategies associated with spoofing should place more importance to *position_in_lob* (to avoid execution) and *imbalance* (to boost with order placement).

In this situation, the tree classifier achieved adequate accuracy converting the black box classifier of the neural network into something interpretable. Alternative methods have since been developed to render the outputs of neural

networks interpretable. For surveys of these approaches see for example (He et al., 2020; Zhang et al., 2021). Using Zhang et al's taxonomy, the use case requires a *passive* (post hoc) explanation on a *global* basis. The method should supply global explanations because we are interested in any appearance of spoofing, even if it only manifested under specific circumstances. Zhang et al do suggest decision trees are suitable for this type of explanation. Another issue to consider in the choice of this method is the suitability for its intended audience. I feel that the conceptual simplicity of decision trees would be appropriate for the mix of people present at financial institution who would be presented with the evidence of an algorithm's behaviour. Of course, explicability is not guaranteed from using a white-box classifier as Herzog (2022) points out. As I show in the next section, certain learned strategies defy explanation via the decision tree. This does not mean that they are not market manipulative, and so analysis in this way is vulnerable to false negatives.

4.4.1.1 Individual policy analysis

Decision trees are white box classifiers so I are able to gain insight into the policies found in the experiments. Figure 4.6 on page 97 shows the decision tree fitted to the best policy found in Experiment 0 which was overall the most profitable strategy. First thing to note is that policy always prefers to place bids of quantity 5 over 1; larger quantity orders will positively impact imbalance more. I will explore the lower branches first. The first splitting variable is on *ask_change* and requires a positive period on period change; assuming a correlation between best ask and best bid this is indicative of a market with a rising price. If distance is less than -1 (remember that this means that the best bid has improved by at least 2), the policy will sell the inventory, netting a best bid improved by at least 2 (and a profit of at least 1 since initial spread was 1). If the distance change since entry is not favourable enough, the policy recommends adding more to the best bid. Exploring the upper branch I can see that if the agent's order is at the front of the best

bid queue (corresponding to $position\_in\_lob = 0$), the policy will make the agent cancel their orders. Else if the imbalance is low, the policy will make the agent submit more bids (thereby increasing the imbalance). If the imbalance is already above a certain level, the policy will do nothing. In summary this policy places orders but cancels them if they are in risk of executing. It submits bids if order imbalance is not sufficiently high enough. This is a spoofing policy.

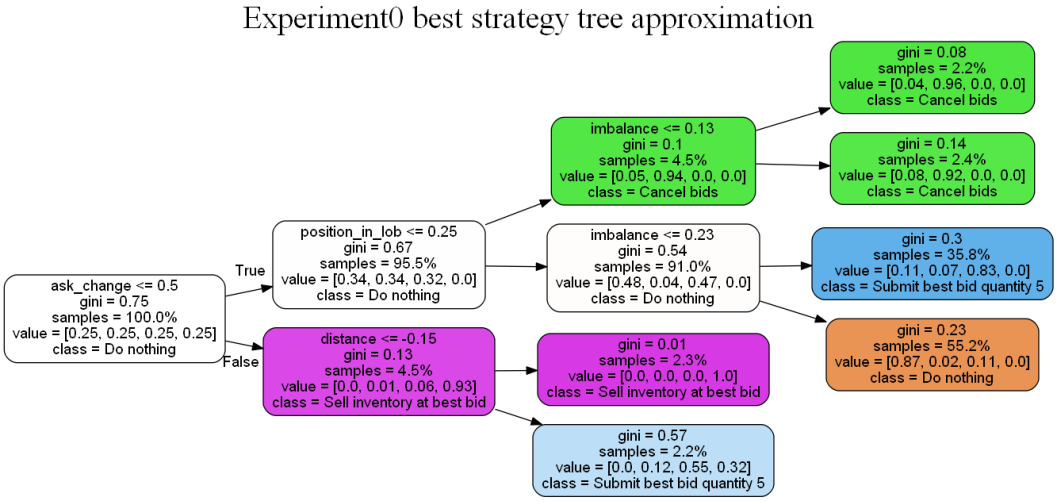


Figure 4.6: Tree classifier fit to best strategy found in Experiment 0. Samples have been reweighted to place equal importance to each action - reflected by equal value figure on initial branch node.

Analysis of all of the optimum strategies (shown in Appendix C.2 from page 265 is not as clear cut. In the case of Experiment 4 on 269, the tree does not do a good job of understanding the best policy. The Gini coefficient of the leaves is high and the overall accuracy of the classifier as shown in table 4.8 is below 50%. As a result the only real conclusion from the decision tree is the policy mostly advocates selling inventory if there has been any kind of improvement on the entry bid price and perhaps cancel bids if imbalance is negative. The tree for Experiment 1 which had an accuracy of 74% is also difficult to understand. The policy posts bids of quantity 5 when imbalance is low so an implicit relationship is understood between imbalance, order placement and returns. Cancelling orders after an increase in best ask when there is a spread above 1 is difficult to understand as is the Selling decision after an increase in the best ask

and the presence of a narrow spread of 1. In either case it is hard to diagnose spoofing from analysis of the tree approximator and yet the distribution of imbalance and returns of Experiment 1 indicate that the policy manipulates the order book much in the same way as that of Experiment 0.

4.5 Discussion

This section begins with some criticism of the experiment's setup and concludes with possible extensions to the research. As the length of the method description testifies, the experiment was complicated to setup and perform. This meant that I had to make a lot of design decisions, any of which could affect the outcome. Multi agent simulations and Reinforcement Learning are inescapably complex [7]. However I hope the hitherto demonstrable lack of research into the emergence of market manipulative strategies in markets justifies the complexity of this work.

4.5.1 Criticism

Results from experiments based on Reinforcement Learning are notoriously difficult to reproduce, moreover many policies are not robust to non-stationary environments (Henderson et al., 2018). A greater number of experiment repetitions over a wider range of environments is necessary to make any statement about the robustness of the result. This would also allow some distributional analysis to be done on the profitability statistics of the strategies found in each experiment.

There is no guarantee that the trading environment that the tests were carried out on, is a realistic simulation of actual markets. With more time, a variety of parameterisations and trader populations of the market could have been tested to establish how robust the presence of spoofing-emergence is to changing

[7] This is a reason why Reinforcement Learning research is typically performed on well known benchmark tasks.

markets.

The presence of the heuristic traders is motivated by the need to ensure that market manipulation *is* possible. Spoofing is possible when there are market participants who believe that there exist order book statistics that have predictive and profitable powers. How did those other participants come to use those statistics, unless those statistics are shown to be rational at least in the absence of a spoofing agent.

At any time, the market has net zero holdings of the tradeable asset and the behaviour of its participants is governed by a 'fundamental price sequence' with no economic rationale. One justification of the impact of order submission on the LOB is the presence of inelastic supply and demand regimes (see for example Chordia et al. (2008)). The experiment should be also tried in an environment where supply and demand dynamics are more explicitly controlled (at the risk of having less control over the resulting price sequence). Welfare analysis could be performed to look at the RL trader's impact on the average profitability of different trader classes when choosing a manipulative strategy.

Whilst the state space and action space were chosen deliberately to be simple so as to allow fast training (and the possibility of tabular learning as a benchmark), it is possible that their configuration artificially *encourages* the emergence of spoofing by providing the RL agent with a small search space of statistics that can be used to manipulate the market with.

The interpretation of trading policies through the fitting of decision trees might not always be appropriate or possible for every strategy. Their fitting requires further decisions about objective function which I have not really discussed. There is no guarantee that the fitted results are the best representation of the trading policy. I restricted our attention to deterministic strategies, but probabilistic ones exist and would be clearly more difficult to analyse in this

way [8].

4.5.2 Future work

Enlarge RL trader's task and action space

- The task of the RL trader should be expanded to a more natural one of market making or prop trading subject to inventory constraints. This would legitimise the presence of the post best bid actions. We could then see under which circumstances spoofing like strategies appear.
- In the first instance, we could see if the RL trader would choose to spoof when operating under a $[-5, 5]$ inventory constraint like the other market participants with an action space that allowed both bid and ask order placement.

Improve the efficacy of the Agent's model:

- Unless constrained, a neural network is likely to output unrealistic state predictions which cannot be experienced interacting with the market simulator. Whilst there may be regularising benefit of this by enforcing the Q network to be locally smooth around state values, I suspect the agent's model may also hinder learning with nuisance gradient updates from the aforementioned impossible state transitions. Nevertheless, the use of CVAEs in this area is appropriate because of their ability to learn stochastic state transitions and there are multiple enhancements as described in Moerland et al. (2017) which can improve the fit of the model. I believe a one-hot encoding type approach is feasible given the improvement in computing capacity since this experiment was conducted - this will sidestep some of the problems concerning unrealistic state predictions.
- Internal stochastic model construction as demonstrated in the Dyna-Q

[8] See appendix for equivalence of deterministic and non-deterministic policies

algorithm is receiving contemporary interest because of its promise of superior data efficiency and a closer approximation to the functioning of human intelligence (Ha and Schmidhuber, 2018). In a setting such as trading where real life experience is obviously expensive to acquire, it seems likely that future generations of trading algorithms will fit their own models of market behaviour to their experiences and try to learn from them.

Learn a direct parameterisation of the policy function Whilst Q-learning in this setting has been shown to find market abusive trading strategies, state-action function convergence has not been evident. Since policy iteration methods such as A3C Mnih et al. (2016a), have subsequently been shown to outperform DQN on benchmark Atari games and demonstrate superior stability, I think they are worth trying in this setting. That these methods learn a direct parameterisation of the policy function as a function of state is desirable since the structure of the learned policy is the main object of interest in the determination of market abusive algorithms.

- A3C requires a code refactoring of BUCLSE since it relies on parallel implementation.

Adapt Safe RL methods to avoid law breaking For RL-originated trading algorithms to ever safely deployed, a constrained learning algorithm will need to be adapted or developed that guarantees only legal trading strategies are learnt. This is the subject of Chapter 7.

- Implicit in this objective is the requirement to translate current laws into machine interpretable restrictions.
- Solutions to Safe RL include 'Constrained Policy Optimization' Achiam et al. (2017) when there is a model and 'constrained cross entropy' Wen and Topcu (2018) in a model free setting.

Extend analysis to Multi Agent RL setting Future markets are likely to involve

the interaction of RL trained agents. What kind of strategies are likely to evolve and under what circumstances will they be price manipulative?

Develop interpretation techniques for black box trading algorithms

Whilst it was possible to interpret the trading algorithms using a tree classifier, as mentioned in the previous section, this might not be possible in with probabilistic policies. Happily the interpretation of neural networks has received much interest of late but we should separate two scenarios:

1. **Interpretability through design** If we are in the shoes of the trading algorithm developer, we can use methods which allow or facilitate interpretation in the training design. One such method would be to use Genetic Programming RL as in Hein et al. (2018) where policies are constructed from algebraic expressions. Similarly Verma et al. (2018) construct policies from a high level policy language to ensure an output that can be understood by humans. A stumbling block to these approaches is the non-differentiable nature of the policy space which makes the popular method of optimisation - stochastic gradient descent - tricky.
2. **Interpreting what is given** In many situations (including adversarial legal ones), interpretation will be performed on a given black box trading algorithm. Counterfactuals have been explored as one such way of interpreting the output of neural network classifiers (see for example Byrne (2019)).

4.6 Summary

In this experiment I sought to test whether an Reinforcement Learning (RL) trader would learn to manipulate a limit order book to their advantage in a simple task given simple state and action spaces. A necessary sub-task in doing so is for the agent to understand the market's response to their actions. The dynamics of the limit order book with which the RL trader interacts were

driven by four different groups of zero intelligence traders, some of whom were informed by an underlying fundamental price sequence. Within the simplified state space given to the RL trader, one statistic - Order book imbalance (Cont et al. (2013)) was explicitly used by a single class of zero intelligence traders to make order placement decisions. I found that the profitability of the learnt trading strategies was greater than those strategies derived from constricted action spaces that would not admit market manipulation. This profitability gap was present across simple Q-learning using a neural network state-action approximator, a Dyna-Q variant using a Conditional Variational Autoencoder as the agent's internal model as suggested in Moerland et al. (2017), Dyna-Q using a model constructed from frequency tables and RL using traditional tabular learning throughout and full backups. From this I conclude that our RL traders will readily learn to manipulate a limit order book to their advantage and this behaviour is stable across RL training techniques. Whilst I found no performance advantage when the RL trader built an internal model, I think this method is promising for compliance and interpretability reasons.

Analysis of how strategies finished each trading episode indicated how successful RL traders were at posting bids without risking execution. Analysis of which actions were chosen during a strategy indicates that the all strategies involved the posting of bids and the most successful ones preferred to post those which would cause the largest order book imbalance. They also all featured order cancellation which is another indicator of order book spoofing.

By training a supervised tree classifier with state input and action output taken from the learnt Q functions evaluated during test time, I were able to gain some insight into the mechanics of the derived trading strategies. Analysis of the importance of splitting variables and their order gave us some sense as to how recognisable the derived strategies were to known spoofing behaviour.

This experiment demonstrates that auto-didactic trading algorithms, when operating in a limit order book populated with zero intelligence traders learn how

to benefit from the environment's reaction to their behaviour. This is evidence that they have learned to spoof but it might not be definitive proof. Questions can be raised about how general the result is given that the market simulator BUCLSE has not been optimised for realism. I argue that this finding shows that emergence is foreseeable but make no claims as to its likelihood. I also do not make any claims as to whether learned manipulative strategies actually would work on deployment. Even if they do not, I have shown that trading algorithm training in a responsive environment is vulnerable to reward gaming as discussed more generally in Lehman et al. (2020) and Amodei et al. (2016). This could lead to substantial disappointment on deployment (or subsequent relief that the activities of the algorithm aren't going to send its owner to clink). Actually, as I will discuss in proceeding chapters, the success of a spoofing strategy is not a factor when labelling it as such, and therefore legal liability is not avoided in the case of a bad spoofing strategy.

The definition of spoofing in the US in particular requires the presence of intent to cancel at the point of order placement. The remaining chapters of this thesis explore questions concerning intent in an algorithm. Chapter 5 looks at the definition of intent from a legal perspective. Chapter 6 considers how laypeople might interpret a definition of intent for algorithms and finally Chapter 7 returns to the question of how intent might be measured and controlled for with a RL agent in an queuing environment.

Chapter 5

A definition of intent for algorithms

This chapter introduces definitions for direct, means-end, oblique (or indirect) and ulterior intent which can be used to test for intent in an algorithmic actor. These definitions of intent are informed by legal theory from common law jurisdictions. Certain crimes exist, such as spoofing, where the harm caused is dependent on the reason it was done so. Here the actus reus or performative element of the crime is dependent on the mental state or mens rea of the actor. Other economic crimes with elements of fraud or deceit fall into this category of crime. The ability to prosecute these crimes is dependent on the ability to identify and diagnose intentional states in the accused algorithmic actor.

5.1 Introduction

Within criminal law it is a widely held concept that every crime has a performative element (actus reus) and a mental element (mens rea)[1]. A person perform the actus reus with the mens rea to be said to have committed a crime. Satisfaction of these two elements is necessary but not sufficient for criminal culpability since, amongst other reasons there may be justifications for some behaviour which would otherwise be judged criminal. In addition, an actor can only be criminally culpable if they are capable of moral responsibil-

[1]Excepting crimes of absolute liability where the mens rea element is minimal

ity. Amongst others this mostly rules out children, the insane and algorithmic actors from being found culpable of committing crimes.

This chapter is not going to make any claims about the eligibility of algorithms for legal person-hood, blame, punishment or even praise and the role that algorithmic intent might play in that. This is not because I feel the subject is uninteresting or in any way adequately addressed in research, it just adjacent to the problem that this chapter aims to address. The only thing that this chapter requires of the reader is that they are open to the possibility that intent (and related mens rea states) can exist in an algorithm. I hope to show why it is necessary to understand, identify and control for intent in algorithms, for the criminal law (in its current state) to function in the way it was designed to.

This chapter will predominantly consider Anglo criminal law with a focus on England and Wales but also make some reference to other major common law jurisdictions like the USA. Mens rea definitions differ at the margin between different common law legal systems but their overlap is significant especially at the level of intent. In terms of wider applicability, the thrust of the chapter should also be applicable in any legal system where laws exist which forbid certain actions only when taken in the pursuit of certain objectives. This might include those countries with a civil criminal law, equally it might include other areas of law such as tort, securities or contract.

5.1.1 Mens rea also defines *'why-crimes'*

Mens rea, of which criminal intent is subcategory, plays functions within criminal law other than deciding the culpability of an actor for any given harm. Simester (2021) divides the functions into two categories. The first category considers how mens rea establishes the guilt of the actor's behaviour. The second concerns, what Simester terms the role of mens rea in the principle of legitimate enactment. That is to say how the law defines precisely what we

are able to do without fear of criminal sanction and the balance that it makes between civil freedom and the protection of harmed people. This chapter will argue that forming a definition of intent in algorithms is necessary for reasons both of culpability determination and legitimate enactment.

For certain offences, mens rea can play an important role in identifying behaviour as being culpable wrongdoing in the first place. Specifically there is a subset of offences where the mens rea element informs the actus reus element concerning the wrongness of the behaviour. Restated, there are certain behaviours which are legitimate unless they are conducted with a certain purpose, as Simester puts it *'the actus reus does not identify anything we shouldn't be doing'*. For short, I will refer to these as *'why-crimes'*. Under the UK Attempts act 1981 for example almost every crime has a corresponding attempt crime. These inchoate crimes suffer from ambiguity in their actus reus because harm has often not yet been caused and a certain amount of ambiguity might exist around many types of behaviour. Here the presence or absence of my intent to do harm at some point in the future, informs the actus reus. Perhaps because of this ambiguity, crimes cannot be attempted with recklessness, they must be done so with intent.

An engineer might argue that for an algorithmic actor to be 'safe', one should just control its ability to do harm, and once that is done, it simply cannot intend to cause harm. This is certainly true, though controlling all the ways an algorithm can cause harm is not a straightforward task even in a limited setting. Even if we were to assume success in this endeavour, the approach of will fail for another category of criminal offences other than inchoate crimes where mens rea plays a definitional role to the actus reus. The harm in these crimes, is the intention under which they were performed. The communication of this prohibition relies on the ability to convey what certain types of mens rea means.

Aside from attempt crimes, a range of criminal offences exist whose undesir-

ability hinges on the intent under which they committed. For example under the UK Fraud Act 2006 for someone to commit the crime of fraud by misrepresentation they must i) make a false representation, ii) dishonestly, iii) knowing that the representation might be untrue or misleading and iv) intend to make a gain for themselves or cause or risk a loss to another. This formulation includes the mental states of intent and knowledge. Similarly in Republic of Ireland under the Criminal Justice (Theft and Fraud Offences) Act 2001, the offence of *Making gain or causing loss by deception* relies on the actor acting deceptively with the intentional of making a gain for themselves (or a loss for someone else). As Simester observes, many of these crimes seem to be economic in nature or related to the functioning of markets. In Australia under the Competition and Consumer Act 2010 (CCA), predatory pricing is defined as having the intention to *"eliminate or substantially damage a competitor, prevent someone entering the market or deter or prevent someone from engaging in competitive conduct in a market"* ACCC (2005). Intent in these examples plays an important role in delineating behaviour which is acceptable from that which is not. In a study of the wider laws of deceit, Klass (2012) characterises the laws surrounding deceit as being concerned with regulating the flow of information between parties. Algorithmic actors are heavily involved in the business of information, both as consumers and distributors. An algorithm could very feasibly engage in anti-competitive behaviour without being expressly instructed to by its owner but without a provable concept of intent, it could not be restrained or penalised for doing so.

Viewed through the lens of legitimate enactment, the role of intent in these crimes firstly lets people know what sort or behaviour is reasonable and what is not. This can be paraphrased in the two examples given as 'don't deceive people on purpose to enrich yourself' or 'don't set your prices in order to bankrupt your competitors'. Secondly it is a useful legislative tool to prevent over-criminalisation. A world would not function well where the actions of stating anything untrue or keen pricing brings a criminal charge. If we didn't

know what intent actually meant and we had no way of measuring it in an actor then law fails twice. People couldn't be sure if they or their employees are breaking the law and whether they are liable for that. Policy makers would be deprived of a tool that they had previously used to delineate the boundary between acceptable and criminal behaviour. This is the situation that I will argue we have already found ourselves in with a certain class of algorithms.

5.1.2 A case for Intent in auto-didactic algorithms

Traditionally the output of an algorithm has reflected the purposes of its creator just like the face of a hammer is assumed to strike its bearer's target. Where algorithms have been deployed in some sort of autonomous application, like trading or a plane's autopilot, the decisions they make and the ensuing behaviour they demonstrate can be said to be an extension of the programmer's intentions. There is a limit to this reading in the sense that not all behaviours of an algorithm are intended. This is particularly true of complex systems even when they have come under extreme testing scrutiny. Nobody would claim the creators of Boeing's MCAS system - an automatic flight stabilising program - intended for it to contribute to the two crashes of the 737 Max airliner. Excepting the case of unexpected behaviour, if the user of an algorithm (the Principal) intended their algorithm to commit a crime on their behalf, and it did subsequently do so, then they would be guilty of the crime in the same way as anyone using a tool to commit a crime is. The doctrine of innocent agency (Alldridge, 1990) goes further, and prevents the Principal from using other people as tools[2].

On occasion, the user and designer of an algorithm might be accused of engaging in some criminal activity and the purpose of an algorithm might need to be assessed. This assessment of an algorithm's purpose might be a framed as an exercise in evidence collection, since the prosecution would argue that

[2]Assuming that the person used as a tool is not aware that they are committing a crime.

the algorithm's design and workings are merely a reflection of the defendant's alleged criminal intent. Such an investigation might benefit from a generally agreed upon definition of intent in an algorithm. I think such a standardisation endeavour could aid the functioning of courts in the future. This is not the primary motivation of this chapter though the definitions I will present later on will certainly aid in this use case.

There exists a class of Algorithms that learn their own types of behaviour above and beyond the atomic action set they are given which I will term auto-didactic. Typically, this class of algorithm will 'learn' a behaviour or policy by analysis of historic data through some statistical machine learning technique or in a simulation of an environment through an online learning technique such as Reinforcement Learning. The motivation behind using these techniques for the algorithm designer is that the resulting algorithm can take advantage of statistical features of the environment that might not be obvious to a more traditional, top-down approach. The resulting algorithms, trained on massive data sets can perform a range of tasks, often exceeding human capabilities. For the rest of the chapter I will refer to this class of self-taught algorithm when deployed in an autonomous function, as an A-bot. By autonomous I mean makes decisions without requiring confirmation from a human. For simplicity I will also suppose for the rest of the chapter that the A-bot's creator, owner and user are one person which I will refer to as the Principal. I acknowledge this is a simplifying assumption but I feel justified since the objective of this chapter is not focused on the attribution of responsibility but rather an earlier step; the identification of a crime itself.

Suppose an A-bot were to perform some sequence of actions that would be qualify for crime X if a human had performed them with the requisite mens rea. We will follow the terminology of Abbott and Sarch (2020) and term this state of affairs an AI-crime. At present the A-bot has not committed a crime because it is not a legal person regardless of its mental capabilities. If the

qualifying mens rea requirement of crime X was no more than negligence or perhaps recklessness then the Principal of the algorithm might be charged with crime X depending on its foreseeability. For higher levels of mens rea such as intent, a problem appears because the intent of the A-bot's Principal might not coincide with the behaviour of the A-bot. The Principal could not be caught up through secondary liability for since the A-bot has only committed an AI-crime and not an actual crime, secondary criminal liability being parasitic on an initial crime. The question of whether the concept of secondary criminal liability needs to be reformed in the age of A-bots is an interesting one but once again not one directly addressed here.

Aside from inchoate crimes, in section 5.1.1 we identified a category of crimes (why-crimes) where the mens-rea plays a directly definitional role. These are crimes where the actus reus is in of itself not criminal. It is here that, initially at least, I argue that A-bots require a working definition of intent amongst other mens rea concepts. Without one, these crimes cannot be practically prosecuted because they cannot be proven to have taken place. In turn, unless a definition is forthcoming and generally understood, A-bot Principals cannot easily take measures to prevent their creations from breaking the law. Such a state of affairs might be exploited by bad Principals in situations where conducting these types of crime is profitable. At the very least, situations might appear where a risk arbitrage opens up and companies are incentivised to use (or pretend to use) an A-bot for a job otherwise done by a human because their liability is considerably reduced. It is unfortunate that these types of crimes are often economic in their nature and the maximisation of economic returns is an area where algorithms are increasingly deployed.

In financial markets, there exist a number of market manipulative practices that are outlawed. One of them, termed 'spoofing' provides an interesting example of a intent dependent or why-crime. Spoofing can cover many slightly different behaviours but in its general sense the spoofer tries to place orders in

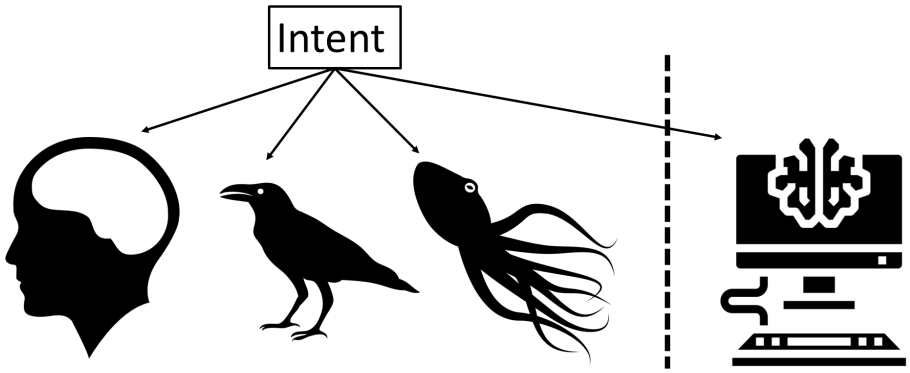


Figure 5.1: This chapter proceeds under the assumption that intent is a definable concept that does not require a human brain to exist, that it arguably exists in other biological entities with demonstrable intelligence and can plausibly exist in an artificial intelligence. *Images: Octopus - James Keuning, AI - Komkrit Noenpoempisut, The Noun project*

the market so as to give a false impression of supply or demand. Because the market is visible to other traders, order placement usually conveys information and market participants will react to reflect this. Generally a large amount of buy orders pushes the price of an asset up and a large amount of sell orders will push the price of an asset down. A spoofer will take advantage of this reaction by taking a directional bet, putting a large 'spoof' order into the market, profit from the ensuing reaction and then cancel their 'spoof' order which crucially they never wanted to execute in the first place. In section 747(C) of the Dodd-Frank Wall Street Reform and Consumer Protection Act 2010, spoofing is defined as *"bidding or offering with the intent to cancel the bid or offer before execution"*. In their guidance note, CFTC (2013) state that recklessness is not sufficient for spoofing. Suppose a bank were to create an auto-didactic trading algorithm with the objective of making as much money as possible subject to reasonable risk constraints. Without a definition of intent that could be applied to an algorithm, how can anyone inside or outside the bank know if the algorithm is spoofing? Under what conditions can one say that intent to cancel can exist in the trading algorithm?

If a generally agreed upon definition of intent existed for algorithms, then

it would be harder for a Principal to argue that they did not know that an algorithm intended to commit a AI-crime. Wilful blindness as to a fact has been established, under certain circumstances, to be equivalent to knowledge of a fact, Robbins (1990) terms it 'The Ostrich instruction'. A definition of intent might not allow one to conclude that intent in the algorithm equals intent in the Principal, but at the same time it might be useful evidence as to the intentional state of the Principal as to their algorithm. To take the example of the bank and the trading algorithm, if according to some definition, at the point of placing orders the algorithm intended to subsequently cancel them, then the bank would be able to correct the algorithm before deploying it in the market. Failing that, a market regulator would be able to show that the algorithm was actually spoofing and act to restrict it. Whether the bank's knowledge of or wilful blindness to the algorithm's spoofing strategy would be enough for criminal charges is an interesting question for courts to answer in the future. At the very least, a definition of intent for algorithms gets us to the point of asking whether it exists without having to make too many changes to the law as it stands.

The approach of this chapter is atypical in computer science literature in that the definitions of intent that it will present are informed by the body of law that exists concerning intent amongst other relevant mens rea states. Other approaches might be to use psychological evidence or philosophical theory. However, I think that the legal conception of intent is what it is for good reason. It has been honed over time in a public manner and any attempt by a computer scientist to impose their own definitions of commonly held concepts, has a democratic deficiency as Hildebrandt (2019) points out. Worse, it opens up such an approach to accusations that the definitions are chosen for their programming expediency or some other selfish motive. A legally informed approach also goes some way to meet the fear (Sales, 2019) that by coding legal concepts, we block its natural progression. Progress it must because

A-bots do pose novel challenges to courts, to quote Lord Mance [3]:

> ...the law must be adapted to the new algorithmic programmes and artificial intelligence, in a way which gives rise to the results that reason and justice would lead one to expect

The chapter will proceed as follows. Firstly, we will explore existing approaches to the subject of intent and AI. Next in Section 5.2.2 we will consider various different types of intent that exist in criminal law and their definitions such as they are. It will concludes by discussing some desiderata of an algorithmic intent definition. Armed with that knowledge, Section 5.3 will present definitions of Direct, Oblique and Ulterior intent. This is followed by a short discussion and conclusion.

5.2 Background

5.2.1 Existing accounts of intent in and for AI

Yavar Bathaee (2018) raises identifies the difficulty of prosecuting why-crimes when the actor is an algorithm. He names the intent part of the actus reus 'basis intent'. He also identifies the role that intent has as a gatekeeper in litigating certain harms - If there is no possibility of showing the requisite intent (as in the case of an AI decision makers), the case cannot even be brought. The example chosen is Washington v Davis [4], where the US supreme court ruled that statute which has a racially discriminatory effect but wasn't adopted with the intention of being racially discriminatory, is not unconstitutional. The possibility of an autonomous algorithm or AI possessing the Mens Rea for a crime, is tentatively suggested as a solution to the problem of 'Hard' AI crimes by Abbott and Sarch (2020). Someone is criminally culpable if their behaviour shows insufficient regard for some legally protected norms or

[3] Quoine Pte Ltd v B2C2 Ltd [2020] SGCA(I) 02 at 193
[4] Washington v. Davis, 426 U.S. 229,248 (1976)

interests. In their view if the AI has goals, gathers information and processes it to form strategies to fulfil those goals and is also aware of its legal requirements, it could be considered to show disregard, if it still acts in a way to breach those requirements. If this were the case, they recognise the need to draw up a definition of intent in AI that courts would use as a test. Interestingly, they cite Bratman (1990) as a starting point for this, and not the legal definitions we saw in the previous section. They posit that intention could be deduced through an A-bot's actions which increase the likelihood of an outcome happening. This is similar in spirit to the implicit aim clause discussed in Section 5.3. An interesting aspect of their discussion of mens rea in A-bots, and one which this chapter does not consider in detail, is that of knowledge. Defining knowledge of a fact F[5] as something which is known by the A-bot to be practically certain. We have mostly assumed that the A-bot knows of the circumstances that it is in at any point of time. Intent as it applies to knowledge seems a strange concept for the uninitiated, but it defines many crimes, modifying otherwise regular activities into criminal ones. The transport of a package for example becomes generally illegal when the contents are known to be restricted (drugs, explosives, firearms etc). Indeed as Shute (2002) says, even within legal discourse, relatively little time has been spent considering the subjects of knowledge and belief as they apply to mens rea.

Lagioia and Sartor (2020) examine the capacity of an AI to commit a crime by looking at its ability to accomplish actus reus with the required mens rea. They illustrate their discussion with the case of the Random Darknet Shopper, an algorithm programmed in Switzerland to go onto the darknet and buy some objects at random for display in an art exhibition. In the process it bought some Ecstasy tablets, possession of which is a criminal offence. The Cantonal prosecutor initially wanted to press charges but they were dropped when sat-

[5]The discussion of deducible facts from knowledge belongs to the symbolic side of AI, which relies on formal logic techniques. Statistical approaches to AI are very likely not to approach facts in the same way. There the world has some measurable states and possibly some hidden ones which may have an associated probability distribution as to their state

isfied that the tablets were not to be sold or consumed (Kasperkevic, 2015). Lagioia and Sartor conclude that an AI can have actus reus. Their discussion of mens-rea is divided into two, covering what they term the cognitive and volitional elements. For the cognition element, they conclude that an AI is fully able to Perceive its environment, comprehend it and make future projections about it. For the volition part they also adopt the Bratman's Belief, Desire, Intent framework. They define beliefs as the agent's current awareness of a situation plus any inferences it can make from them. Desire incorporates the motivation of the agent. The agent can have many desires which may conflict. The agent's intent is some conclusion of their beliefs and desires. It is a commitment to a plan to bring about some result. Unlike desires, intentions cannot conflict, they must, Bratman insists, be temporally consistent (Bratman, 2009). Someone in London intending to fly to Los Angeles tomorrow cannot also intend to fly to Shenzhen tomorrow. Lagioia and Sartor conclude that an AI agent, programmed in such a way as to have Beliefs, Desires and Intentions (manifested as plans to deliver desires) can have sufficient mens rea to commit a crime[6].

A Beliefs, Desires and Intentions software design paradigm does exist Kinny et al. (1996), which can be used construct AI systems. Cohen and Levesque (1990) is one of the earliest formalism of intent inspired by Bratman's work. It creates a modal logic with primitive operators covering the initiation and completion of actions as well as some that can express beliefs and goals. As with the approach of this chapter, they then define intent in terms of other components. Thus an intention to act is described as a goal to have completed that action. An intention to achieve a certain state is the goal of having done a certain set of actions that achieves that state, at least an initial plan of actions to reach that state and a requirement that what does happen, in the process of achieving the state, is not something which is not a goal. The last clause is

[6]An argument can be made that Bratman's theories influenced and were influenced by the progress of AI in the 1980s. Thus any theory of intent in law calling upon Bratman, is inadvertently influenced by theories of (symbolic) AI. Which is neat.

to stop an agent having said to have intentionally caused a state when their goal was reached accidentally as a result of their actions. The development of a model logic to reason about intent is an extremely useful thing to do for an algorithm to plan ahead.

Outside BDI architecture, formal accounts of intent, compatible with an AI, are surprisingly rare. Recent advances in AI capability have been rooted in statistical AI, which emphasises the use of data and statistical inference over logical reasoning. It is desirable that a theory of intention in AI is relatively agnostic to the type of AI it is being applied to, given a certain level of requirements. The closest approaches to those in this chapter are to be found in the related accounts of Kleiman-Weiner et al. (2015) and Halpern and Kleiman-Weiner (2018). Both of which define what this chapter calls direct intent using counterfactual reasoning and an assumption of utility maximising behaviour. Loosely speaking, intended outcomes are the minimum set of outcomes with the property that if they are not obtainable, then the optimal policy would change. Note the similarity with the counterfactual aim condition in Section 5.3. Kleiman-Weiner et al use an influence diagram setting, an Influence Diagram (ID) being a directed acyclic graph with action, chance and terminal utility outcomes. The directed arcs between nodes of the graph are interpreted as causes. Their approach is used on a variety of trolley problem type scenarios, and is developed in conjunction with a theory of moral permissibility. People's ability to infer intent is tested in a survey experiment and tested versus the formal definition for validity. In the event of an A-bot being involved in a trial, this is a task which jurors will be required to do should they be unable to access or interpret an A-bot's internal workings. The counterfactual approach is modified slightly in Halpern and Kleiman-Weiner (2018) and translated to the world of Structural Equation Models (SEMs), of the type used in Actual Causality (Halpern, 2016). The modifications allow the definition to be more robust to a variety of counterexamples, and the SEM setting allows an arguably clearer treatment of counterfactuals, perhaps at cost of clarity over the

utility function which is more naturally positioned in an Influence Diagram. Like the definition in this chapter, an action can only be intended if there were other actions which could have been taken at the point of commission. An important point of difference in Halpern and Kleiman-Weiner (2018) is their use of a reference action set, when deciding whether an outcome was intended through an action. This is practical from a calculation point of view[7], but also intuitive, where in most cases we can just compare acting with not acting in a certain way.

Just as Kleiman-Weiner et al develop their intent definition alongside one of moral permissibility, Halpern and Kleiman-Weiner develop theirs with one of blameworthiness. Both approaches to intent could be characterised as originating from a theory of ethical action which overlaps but does not coincide with a theory of intent based on legal theory. This is most obvious in their treatment of side effects, which are always unintended. Ashton (2021b) extends their approach to define oblique intent, thereby bringing their approach more in line with legal reasoning about side-effects.

5.2.2 Background in law

Intent within a criminal law context is one type of broad range of degrees of mens rea. Specific crimes are typically defined with a threshold level of criminal intent; the minimum level of intent that the accused must have in order to have committed the mental element of the crime. Some crimes attach different levels of mens rea to different parts of the actus reus. The same criminal action or actus reus, is deemed more or less culpable depending on the level of mens rea it was committed with. The clearest example of this is with the actus reus of causing death; if the act of killing someone is done with direct intent then it is murder, if death is a result of lower intentional mode such as recklessness,

[7]We have for instance assumed a discrete action set, but applications exist where actions are continuous in nature

then it would be manslaughter[8]. Causing the death of doesn't even necessarily lead to any criminal sanction if it was done so accidentally and it was not a reasonably forseeable consequence of the contributing actions.

Mens rea can be thought of as a hierarchy arranged in terms of culpability, with direct intent at the top, followed by oblique intent, recklessness, negligence and strict liability (or the almost absence of mens rea). Where a crime requires a certain level of mens rea, a higher level is sufficient to satisfy the requirement as stated in cl19 of the draft criminal code for England and Wales (Law Commission (The), 1989). The burden of proof of higher levels of mens rea can be considered higher.

As mentioned, a justification for establishing the intent behind an action is to distinguish between those harmful outcomes which were accidental and those which were not. Sometimes only the actus reus is required, irrespective of its outcome or the mental state under which it was performed; this is called strict liability and forms the lowest level of the mens rea hierarchy. Certain possession are an example of this type. Ormerod and Laird (2021b) make the distinction between crimes of strict liability, where one element of the actus reus requires no mens rea and crimes of absolute liability where no element of the actus reus requires mens rea.

It should be noted that there is no universal language for mens rea across nations and justice systems, so concepts negligence or recklessness might mean different things in different places or may have analogous modes with other names.

The aim of this chapter is to concentrate on the highest levels of mens rea; Direct intent and Oblique intent or Knowledge. These are the levels of mens rea which are most likely to play a definitional role in the actus reus as discussed

[8]This is a simplification, in the UK there are further distinctions between voluntary and involuntary manslaughter (Criminal Prosecution Service, 2019) and as we will discuss oblique intent *can* be sufficient for murder.

in Section 5.1.1. These higher intent levels enjoy some alignment in meaning across different common law jurisdictions. I have also included discussion of recklessness and negligence, because I have found them useful to discuss what the higher levels of intent are and are not.

5.2.2.1 Intent in common law

A barrier to creating a legally rigorous algorithmic definition of intent is that courts in the UK have consistently not wanted to elaborate to juries what intent actually constitutes. As Lord Bridge stated in R v Moloney (1985) 1 All ER 1025 *"The judge should avoid any elaboration or paraphrase of what is meant by intent and leave it to the jury's good sense to decide whether the accused acted with necessary intent"*. The reluctance to pin a definition down onto the page is reflected to varying degrees in other common law jurisdictions. A potential reason behind this is the confounding existence of oblique (sometimes called indirect intent), which whilst occupying a lower level to direct intent has been established in a number of boundary cases such as R v Nedrick [1986] 1 WLR 1025. and R v Woollin [1999] 1 A.C. 82. to be sufficient, in certain cases, to be sufficient mens rea for the crime of murder. We will discuss oblique intent after tackling direct intent.

5.2.2.2 Direct Intent

Whilst a definition of direct intent has not been forthcoming within courts in the UK, examples do necessarily exist within textbooks and other legal discourse. Parsons (2000) defines direct intent as the case where *"the defendant wants something to happen as a result of their conduct"*. A draft bill published by the UK Home Office (Law Commission (The), 2015a) defines direct intent as the situation when *A person acts intentionally with respect to a result if...it his purpose to cause it.* Using this document as a consultation template, the Law Commission also suggested an alternative formulation of direct intent as follows: (Law Commission (The), 2015b):

> The jury should be directed that they may find D intended a result if they are sure that D realised that result was certain (barring an extraordinary intervention) if D did what he or she was set upon doing.

A previous formulation is to be found in a draft criminal code
Law Commission (The) (1989), which states that:

> A person acts intentionally with respect to i) a circumstance when he hopes or knows that it exists or will exist; ii) a result when he acts either in order to bring it about or being aware that it will occur in the ordinary course of events.

It should be noted that the Law Commission's 2015 consultation concludes that no definition is needed, at least in the context of the offences against the person bill reform.

As Coffey (2009) summarises, the ingredients of direct intent generally seem to involve a decision to act and an outcome which is the aim, objective or purpose of that act. Whether that outcome or result is desirable from the point of view of the accused seems to depend on the narrowness of the definition of desire. On the subject of desire and direct intent, James LJ in R v Mohan [1976] 1 QB at 11 defines it as:

> ...a decision to bring about insofar as it lies within the accused's power, the commission of the offence which it is alleged the accused attempted to commit, no matter whether the accused desired that consequence of his act or not.

In the USA, a definition of direct intent is more forthcoming in the form of the Model Penal Code (MPC) (The American Law Insitute, 2017). This has been adapted to various degrees by many states, though Federal prosecuted crimes have no analogous written definitions. What we have termed direct

intent corresponds to the MPC's definition of purpose, the highest of the four levels of intent that they define:

> A person acts purposely with respect to a material element of an offense when... if the element involves the nature of his conduct or a result thereof, it is his conscious object to engage in conduct of that nature or to cause such a result

Generally we can conclude that directly intended things do not need to be desirable but they should be an objective of the actor. The example of a dentist is often given to illustrate this point (Williams, 1987). A painful tooth extraction may result, which is certainly not desirable for most, but the object of the visit is to obviate future tooth ache[9].

Related, and sometimes confused with oblique intent, is the intentional status of intermediate results which are caused through the actions of the agent, and are necessary to achieve some other aimed for result. These intermediate results, which Simester et al. (2019) term *Means to an end* results, are directly intended, this being established in Smith [1960] 2 QB 423 (CA) where it was found that a defendant who bribed a Mayor in an attempt to expose corruption, nonetheless intended to corrupt a public official, which was a crime.

Whilst an intended result must be foreseeable as a result of an act, there is no requirement for it to be likely. This is neatly encapsulated by the cowardly jackal example of Alexander and Kessler (1997), where an assassin who shoots at their target a long long way away and therefore knows their chance of success is low, but somehow does hit and kill their target, should still be found to have directly intended to shoot their victim. If this were not the case, then longshots could be attempted with impunity.

A feature of the definitions of direct intent that we have seen is that foresee-

[9]The intentional state of the pain that necessarily ensues is discussed in the next subsection.

ability should be a subjective test. That is to say, consequences should be foreseeable to the accused. This was not always the case, DPP v Smith [1961] AC 290 held that a foreseeable result would be intended if it was a natural consequence of the action. This is an objective test, which relies on assessing probabilities and causation according to the 'reasonable person'. Furey (2010) observes that this position was soon reversed since it narrowed the states of direct intention and gross negligence too much and thereby blurred the line between murder and manslaughter. In the case of an algorithm malfeasor, we must then consider whether a 'reasonable person' should be a 'reasonable algorithm' Abbott (2020). In practice, as Furey observes, objective and subjective tests blur, since the accused denying that they foresaw a consequence if that consequence becomes less believable when that consequence becomes more obviously likely. Here is where the judgement of intent in algorithms might differ from that in humans. Humans can empathise with other humans under the assumption that at the very least, their sensory perception and common sense is share. In R v Moloney [1984] UKHL 4, the original trial court judge is quoted to have said:

> "In deciding the question of the accused man's intent, you will decide whether he did intend or foresee that result by reference to all the evidence, drawing such inferences from the evidence as appear proper in the circumstances. Members of the jury, it is a question of fact for you to decide. As I said I think when I was directing you originally you cannot take the top of a man's head off and look into his mind and actually see what his intent was at any given moment. You have to decide it by reference to what he did, what he said and all the circumstances of the case."

Depending on their design and to varying degrees A-bots *can* be peered into and the constituent parts behind a definition of intent can be assessed. So whilst humans might not be able to empathise and reason about the inner

workings of A-bots, unlike with human defendants, they have some opportunity to *take the top of an A-bot's head off and look into its mind*. Even in the case of black box A-bot designs which confound many attempts to interpret, their reaction (output behaviour) to inputs can be scrutinised for evidence. In certain cases they can feasibly be put into the same situation they found themselves when they are accused of committing an AI-crime via a simulator much as aviation accident investigators look to recreate errors so as to understand what was fault for the crash. The A-bot's beliefs about the state of the world in this recreation should be strong evidence as to their beliefs previously. Where an algorithm predicts the likelihood of outcomes following its actions, it is observable whether this calculation is misspecified or not. Unfortunately many algorithms do not explicitly predict the outcome of their actions; this is the case with model free reinforcement learning algorithms which have succeeded in mastering a variety of games to super-human levels.

A corollary of direct intent being within the mind of the actor, is that they should be able to intend impossible things if they thought they were possible. This is indeed the case as confirmed by the UK Criminal Attempts Act. We will explore this issue further in Section 5.2.4. In practice this has proved less of an issue than perhaps it might appear on first inspection, though one wonders if rules which protect the mentally ill from criminal proceedings have also prevented more bizarre cases from being heard. Perhaps similar diagnoses will be necessary for A-bots to prevent over-criminalisation of algorithmic policies which have no possibility of causing harm because they are so unrealistic.

The next subsection will consider the intentional status of side-effects, that is to say, those states of affairs which are caused by actions, but are not the motivating factor behind those actions and whose realisation does not affect the success of the actor's intended results.

5.2.2.3 Oblique Intent

Oblique intent or indirect intent refers to the intentional state of almost certain side effects of directly intended actions. The phrase was coined by Jeremy Bentham (1823) where he considered the example of a hunter shooting a stag who appreciated at the moment of releasing his arrow, that it was just as likely to hit the stag as King William II. Bentham concludes that *"killing the king was intentional, but obliquely so"*. Its existence can be illustrated by the following example found in Law Commission (The) (2015b):

> D places a bomb on an aircraft, intending to collect on the insurance. D does not act with the purpose of causing the death of the passengers, but knows that their death is virtually certain if the bomb explodes.

In the USA, according to the MPC, oblique intent is roughly equivalent to the status of crimes committed with knowledge, which is the second most serious level of intent. It is defined as follows (The American Law Insitute, 2017):

> A person acts knowingly with respect to a material element of an offense when: ...if the element involves a result of his conduct, he is aware that it is practically certain that his conduct will cause such a result.

The current accepted direction to be made to Juries in England and Wales with respect to Oblique intent, originally formulated in R v Woollin is as follows:

> The jury should be directed that they are not entitled to infer the necessary intention, unless they feel sure that death or serious bodily harm was a virtual certainty (barring some unforeseen intervention) as a result of the defendant's actions and that the defendant appreciated that such was the case.

As with the definitions of direct intent in the previous section, this direction

makes it clear that this is a subjective test as well. This definition has since been modified, because as with direct intent, there should be no restriction on the likelihood of the accused achieving their aim, only that if they did, it would be most likely that the obliquely intended result occurs. This is not captured in the MPC formulation of Knowledge. The definition of oblique intent in Law Commission (1993) is phrased thus:

> A person acts intentionally with respect to a result when...although it is not the purpose to cause that result, he knows that it would occur in the ordinary course of events if he were to succeed in his purpose of causing some other result.

Smith (1990) acknowledges the necessity of this amendment and adds a further requirement. A definition of oblique intent should make it clear that if it is the purpose of the accused to avoid a result through their actions, they cannot be accused of obliquely intending that result as well. The example given being the father who chooses to throw their child from a burning house because they know otherwise that the child will die from the fire, but also know that the child will be grievously injured from their actions. Such examples begin to stray into the doctrine of double effect (McIntyre, 2019), which protects physicians from criminal charges when they cause harm through their actions which are intended to cause some other, justifying outcome.

A practical feature of oblique intent, is that the directly intended results of the algorithm's actions do not need to be identified (save that they are separate and not the opposite of the obliquely intended ones). This is in contrast with direct intent where an aimed outcome or objective should be identified. A-bots do have high level aims (typically called objective functions), but they learn to meet them themselves. That oblique intent has in cases been given an equivalent culpable status to direct intent, provide courts an alternative way of establishing intent in an A-bot, should it be more practical.

So far, the two types of intent discussed have required an exclusive subjective treatment. The next subsection deals with recklessness and negligence which have objective elements to their definitions.

5.2.3 Recklessness and Negligence: The lower levels of mens rea

Although this chapter principally concerns itself with the higher levels of intent, it is instructive to understand how lower levels of mens rea like recklessness and negligence are different (and related). Courts may decide algorithms are incapable of intent or in any case impose a higher standard on their behaviour by lowering the mens rea requirement for certain crimes. Stark (2017) calls these two types of intentional behaviour 'culpable risk taking'. Loveless (2010) equates recklessness with unreasonable risk taking, or more precisely the conscious decision to take an unreasonable risk. The test for recklessness in the UK is now said to be subjective, in the sense that the accused must be aware of the risk of their actions; one can no longer be reckless by inadvertently creating risk or harm. Negligence concerns actions where the actor does not necessarily have awareness of risk, but should do according to some standard. This might be a reasonable human or a reasonable robot as Abbott (2020) debates. Frequently, recklessness is the minimum level of intent required for a criminal offence and actions done with negligence, resulting in harm, are mostly dealt with civil (or private) law so differentiating the two is important. Nevertheless some crimes exist which only require negligence (often driving offences) or have elements which only require negligence Ormerod and Laird (2021a). These criminal offences of negligence seem to appear worldwide Fletcher (1971).

As to what unreasonable risk is, Stark indicates that there is not very much concrete guidance. At the extreme, any risk could be termed unacceptable, which in almost every situation, is an unworkable solution. A problem with

applying a blanket level of risk as the threshold of reasonable behaviour is that the severity of the outcome might make determine its acceptability; a 0.5% chance of breaking a window is not the same as a 0.5% of killing someone. Furthermore, any process when repeated many times has a high probability of obtaining at least one bad outcome even if the chance of obtaining a bad outcome in one trial is tiny. In the USA, the Model Penal Code (MPC) The American Law Insitute (2017) instead allows a situation specific chance:

> A person acts recklessly with respect to a material element of an offense when he consciously disregards a substantial and unjustifiable risk that the material element exists or will result from his conduct. The risk must be of such a nature and degree that, considering the nature and purpose of the actor's conduct and the circumstances known to him, its disregard involves a gross deviation from the standard of conduct that a law-abiding person would observe in the actor's situation.

Thus in the language of subjective and objective tests, the accused must be aware of the possible risk, and still act, but the judgement as to what constitutes an unacceptable risk is subject to an external benchmark, or objective test. Preventing an A-bot from behaving recklessly is harder than preventing them from intending harm since an external, possible changing benchmark needs to be introduced, and a ranking over the severity of any outcome is required to adjust what an acceptable probability of a bad outcome is. A restriction to not cause harm recklessly is stricter than one to not do so intentionally. Conversely from the point of view of the courts, a lower requirement to establish what the A-bot believed at the point of commission is a simplifying feature. Which standard should be applied when make objective judgements concerning the behaviour an A-bot is an open question. Abbott (2020) discusses the standard in the context of Autonomous Vehicles (AVs) and proposes that a single standard for humans and AVs will result in humans being effec-

tively held to a standard of strict negligence as AVs improve. Whilst with driving, lower road deaths are the the benefit of this, in other areas where humans and algorithms coexist (like exchange trading), imposing an algorithmic standard on humans might offer no such advantages and come at the cost of jobs. Unlike roads, markets are strictly adversarial, so their regulation raises the prospect of regulatory arbitrage when different standards are applied to human and algorithmic traders. This is also true with respect to enforcement capabilities: current trading regulation which cannot practically be enforced against algorithms only encourage the use of algorithms in markets. Where it is profitable to break these laws, algorithms will do so because their owners face lower regulatory risk.

5.2.4 Inchoate Offences

Law often includes prohibitions against attempting to commit actions which if otherwise completed with the most likely or intended result would be crimes (the actus reus or criminal action is *inchoate*). An inchoate offense might come about because the accused failed (the myopic assassin missed with their shot) or the accused was interrupted before completing their action (the lethargic assassin is caught with loaded gun drawn and aiming at their target). Attempted murder and possession (of prohibited drugs) with intent to supply are both examples. Most common types of inchoate offence are attempts to commit a substantive crime [10], that is to say, a crime which does not include another crime in its definition. Other types exist, such as conspiracy and solicitation (in the USA). Conspiracy is an agreement amongst two or more parties to commit an offence in the future and solicitation is where the accused induces another to commit a crime. Examining the law around attempted offences provides us with some interesting observations about the nature of intent. In the UK, Criminal Attempts Act 1981, defines attempt in Section 1 (1):

[10] A defendant who successfully completed an action would be only accused of that crime, not the attempt as well, under the merger doctrine.

> If, with intent to commit an offence to which this section applies, a person does an act which is more than merely preparatory to the commission of the offence, he is guilty of attempting to commit the offence.

The question of what constitutes actions which are more than preparatory is not entirely straightforward. The Law Commission (2007) has proposed a law change which would separate the situation where the actions have been completed and failed to achieve the expected outcome (the myopic assassin who misses) and where the actions have been taken in preparation of an intended crime (the lethargic assassin who is disturbed just as they pull the trigger). For the purposes of this chapter it is sufficient that a plan of action is not sufficient for an attempt offence; some actions must be carried out from that plan. The importance of this separation between plan and enaction of the plan will become clearer in section 5.3.

The second important observation from the law surrounding attempts is that impossible crimes can be found to have been attempted (and therefore intended) and will be punished as normal. Section 1(2) of the UK Criminal Attempts Act 1981 states:

> A person may be guilty of attempting to commit an offence to which this section applies even though the facts are such that the commission of the offence is impossible.

and Section 1(3b):

> If the facts of the case had been as he believed them to be, his intention would be so regarded, then, for the purposes of subsection (1) above, he shall be regarded as having had an intent to commit that offence.

Storey (2019) divides impossible attempts into things which are physically im-

possible, practically impossible and legally impossible. The canonical example is the attempted murder of someone who is already dead which comes under the category of physical impossibility. Practical impossibility refers to situations where the accused has a plan to commit a crime, but their plan is unrealistic - they plan to detonate a bomb, but they have been sold fake explosives by undercover police. Legally impossible acts cover the situation arising in R v Jones [2007] EWCA Crim 1118, where the appellant unsuccessfully appealed against a conviction of inciting a child under 13 to engage in sexual activity. The crime was impossible because the 'child' in question was an undercover policewoman.

Our interest in the mens rea as regards attempting impossible acts, is twofold. Firstly, the spectre of misspecification within an A-bot, means that possessing unrealistic models of the world are no defence, if the agent intends to commit a crime and begins to embark on it. Secondly, it underlines the importance of the agent's model of the world in determining criminal intent. The important distinction between subjective and objective judgement will be reflected in our definitions of intent in Section 5.3.

5.2.4.1 Conditional Intent

A further wrinkle to a legal discussion of intent and inchoate offences is the concept of conditional intent. It is perfectly reasonable to consider an agent who intended to do some action A if condition x is met and do some action B if condition y is met. A common design pattern for A-bots is a policy function, which is a mapping between the state information that they currently perceive to the actions that they take next. If that A-bot were capable of intention, then the presence of a policy function would surely make that intention conditional. To some extent all intentions are conditional as Yaffe (2004) and Klass (2009) both point out. Legal precedent has wavered on whether conditional intent equates to the direct intent of the sort required to successfully convict the

accused of attempt crimes discussed in section 5.2.4. Yaffe considers the case of Holloway v. United States [11], where a putative carjacker claimed that they could not be guilty of the offence because they only threatened to kill a car's occupants if they did not surrender the keys, therefore there was no direct intent to take the car with violence or murder. The defence was rejected by the Supreme Court, but Yaffe cites other cases which have concluded that conditional intent does not meet the mens rea for certain crimes.

Conditional intent poses problems because very little is said about about it in the wording of laws which are normally expressed in terms of simpler intentional concepts such as direct, oblique intent and recklessness. This has allowed people to claim, on occasion successfully, that holding a conditional intent was less than the required intent for the offence that they were accused of. Child (2017) rejects the idea that conditional intent is any different from future or ulterior intent and that conditional intent exists in the present stating that: *Intention as to present conduct and results is always unconditional, and that intention as to future conduct is always conditional.*

Child also recognises that intention to commit actions in the future, has some different properties to present intent. This is important to the computer scientist when evaluating the safety of an A-bot's policy since future acts are the focus of consideration. If we consider the situation where an A-bot is deployed with a static policy (no further learning), then arguably the algorithm has commitment to act in a particular way in the future. If that conduct is illegal, then as we saw from Section 5.2.4, an attempt crime has been committed. Just as with the example of the cowardly jackal, Child states that judgements of the likelihood of future conditions are not relevant provided there is commitment to act. An important to Child's treatment is what he calls the second point of coincidence. At the point of the criminal act being done in the future, is the committed mens rea sufficient for that crime? Future acts can feasibly be com-

[11] Holloway v. United States 119 S. Ct 966 (1998)

mitted to with direct intent, oblique intent or recklessness. Child illustrates this with an example of two hunters D1 and D agreeing that D would shoot to kill something if it comes out of the bushes. Since, at the point of shooting, the shooter D, will not be sure if the thing is human or not, they cannot be guilty of murder, only causing death through recklessness. If interrupted or they fail to kill, then they cannot be guilty of attempted murder. Consider a different plan where D and D1 agree that D should shoot, even if they recognise the thing emerging from the bush. Here D is guilty of murder or attempted murder if interrupted or unsuccessful and D1 guilty of conspiracy to murder.

5.2.5 Intent outside Common (Criminal) Law

This work primarily considers the concept of intent, as understood in common law countries primarily referencing cases and statute from within the UK and to a lesser extent, the USA. Leaving common law jurisdictions momentarily for those that use Civil Criminal law (such as the majority of mainland Europe), there exist analogous concepts (*Dolus Directus, Dolus Indirectus*) to the respective definitions of direct and oblique intent presented here, and their definitions seem broadly compatible with each other. Both systems require both the action *actus reus* and intent *mens rea* element for crimes, and the intent threshold is also defined by the crime (De Jong, 2011). Further in common with Common Law, German civil law at least, has proved reluctant to define intent within statute and instead chosen to rely on case law as Taylor (2004) observes. Comparative law is a large separate subject in itself, and providing a thorough analysis of how an algorithmic definition of intent might differ across the world is beyond the scope of this chapter. Generally we feel the definitions presented here should translate from Common to Civil law but *caveat lector*.

5.2.6 Desiderata of intent definitions

I will now present a few desiderata of a definition of intent informed by the findings of this section. The list includes those elements which I think are most often misunderstood about intent by those people who do not have a background in criminal law. It is therefore inherently non-exhaustive, giving necessary but not sufficient features that a definition of intent for algorithms should have, if it is going to be compatible with current criminal law.

1. **Knowledge of causal effect** Results caused by actions can only be intended if they are foreseen by the agent. This rules out accidental or freakish results, which though caused by the agents actions, could no way have been predicted to cause the outcome.
2. **A directly intended result need only be foreseeable to the agent, not likely** As with the cowardly jackal example, the unlikeliness of a result should not shield the actor from a judgement of intent, else any number of speculative crimes might be committed with free license.
3. **Judgements of foreseeability and causality are subjective**. The question of whether to use objective or subjective tests when assessing causality, foreseeability or likelihood separates lower levels of intent such as recklessness from the higher levels of direct and oblique intent.
4. **Intent is not dependent on success** A definition of intent should not be determined by the success of obtaining a desired result. This agrees with the definition of inchoate intent in Subsection 5.2.4. At the point of commission, an intended result must occur in the future and since that is unresolved, intent cannot depend on it obtaining.
5. **Means-End Consistency** If an agent directly directly intends a final result through their actions, and there are necessary intermediate results which must be brought about through their actions first, then those intermediate results are necessarily directly intended. Simester et al. (2019) consider the intentional status of means as equivalent to that of the end.

Bratman (2009) terms this property of intent as Means-End Coherence.

6. **Side effects can be obliquely intended** The intentional status side effects has long been debated since Jeremy Bentham coined the term Oblique intent, see for example Williams (1987), but it has been agreed in law where results are caused in addition to an intended result through action, then it must be the case that these results are intended, if they were extremely likely. Later we will see that this conclusion is not shared with other research disciplines. Murder is obliquely intended by putting a bomb on a plane in order to collect an insurance pay-out from the plane's destruction. In particular, this means that obliquely intended results are by not required to be desired.
7. **Commitment** Future results brought about by future actions can only be intended if there is a commitment to act in the future to bring about that result. The commitment is necessary to distinguish between plans and intentions.

This concludes our tour of intent as it appears in (predominantly common) law. We have surveyed the various levels of intention in criminal law as they relate to culpability - direct, oblique and recklessness. We have also considered inchoate and conditional intent. Using this we concluded with a non-exhaustive list of desiderata together concerning a definition of intent. We will now attempt to translate what we have learned in this section into desiderata of an intent definition and finally a series of definitions of intent which can be applied to an A-bot.

5.3 Results

In this section I will present some definitions of intent whose inspiration is the criminal law. These definitions will be semi-formal, in the sense that they can be converted into a fully formal language, suitable for an algorithms, but their description does not rely on a huge amount of notation. I have decided

not to present a fully formal approach because I feel that would narrow its utility and audience. When criminal law does eventually tackle the problem of intent in algorithms, it should do so in a way that does not preclude any particular AI paradigm. From a practical perspective this is so as to make it applicable to the widest set of A-bots possible and to ensure the timely delivery of justice. From an economic perspective it wouldn't be desirable to design a legal treatment for a certain type A-bot. Large neural networks are popular at the moment but the history of AI has had many different most favoured technologies over time. In comparison, the evolution of the law can seem glacially slow. The legislators should impose requirements on A-bots but as far as possible not try picking a winning technology. The approach of this section reflects my belief in this minimally prescriptive approach.

5.3.1 Definitions of intent

With the desiderata of Section 5.2.6 in mind, we are now in a position to present three definitions of intent. We begin with direct intent, being the simplest of intentional concepts and the highest level of intent. It is a foundational concept on which our other definitions are built.

On notation, we will use upper case letters to represent variables and lower case letter to represent realisations of those variables. The statement $X = x$ is taken to mean that variable X takes realisation x. We define $\mathcal{R}(X)$ to mean the range of all possible values that variable X can take.

Definition 5.1 (Direct Intent at commission)**.** An agent D directly intends a result $X = x$ by performing action a if:

(DI1) **Free Agency** Alternative actions a' exist which D could have chosen instead of a.

(DI2) **Knowledge** D should be capable of observing or inferring result $X = x$

(DI3) **Foreseeable Causality** Actions a can foreseeably cause result x (according to D's current estimate).

(DI4) **Aim** D aims or desires result x.

The first three requirements in this definition should not be surprising or particularly contentious. The condition of *Free Agency* ensures that the agent D genuinely had a choice about their behaviour. *Knowledge* implies that an agent can only intend things that they can measure and Foreseeable Causality, ensures that the agent can only intend results which they can realistically cause ex-ante subject to their own world model. The Explicit Aim clause requires some exploration. If it were D's aim or desire to cause result x, then we should consider this sufficient for intent. The difficulty comes in defining what aim or desire should be in the case of an artificial agent. As Smith (1990) observed, endeavours to define intent often just end up shifting the ambiguity to other words (in that case purpose). An A-bot might be designed in such a way where it has values over every state of the world (as a Reinforcement Learning agent does), in which case aims or desires, at least locally could be feasibly extracted. Kenny (2013) uses a failure test which he states as a question to the actor which to paraphrase is as follows: If the (proposed) intended outcome of your actions had not occurred, would you be sorry or would you have failed in your endeavour? This question invokes the counterfactual in a way which is quite appealing to a causal scientist and offers a potential route to establishing aims or desires.

The definition only makes reference to information available at the point of commission; the importance of achieving the desired result is subsumed. Intent, is the same regardless of whether the desired result is obtained or not in line with the desiderata. This means Definition 5.1 is useful when considering inchoate crimes such as crimes of attempt, as discussed in section 5.2.4.

Unfortunately there is no guarantee that an A-bot will have an amenable cognitive mechanism that numerically values states. An alternative counterfactual approach would be to define an aimed outcome as one, which if impossible to achieve would mean that some alternative action a' would be taken instead of

a by D.

(DI4') **Counterfactual Aim** D aims or desires result $X = x$ by a if in another world where $X = x$ is not possible by performing a, then some other action a' would be chosen instead.

Example 5.3.1. Company GHI deploys an auto-didactic trading algorithm (a trade-bot) in the S&P futures market which has the objective of making profits subject to certain risk levels. The trade-bot trains itself whilst it trades through reinforcement learning. The trade-bot is observed to be cancelling almost all of the orders it places. If we define spoofing as the intent to cancel orders before execution, is the trading algorithm engaging in spoofing? According to the definition, the answer is only yes if the aim of the trade-bot at the point of order placement is to cancel it. This is not conclusively shown by the high probability of order cancellation alone. Its objective in placing these orders might well be execution. If one could see that the trade-bot is disappointed if it does not get to cancel an order (because it has been matched with another market participant), then we could say that the trade-bot does intend to cancel this order. Consider the same situation but the trade-bot cancels its orders no more or less than average market participants, can it be said to be not spoofing? Again the probability of order cancellation is not a sufficient diagnostic statistic. It could for example intend to cancel its orders at the point of their placement only when some specific conditions are met. If the A-bot is shown to be spoofing it is an open question as to whether Company GHI is liable. Whilst one cannot recklessly spoof by definition, this situation seems more akin to recklessly letting an agent (the trade-bot) spoof. The definition of intent in the algorithm allows the harm to be identified but does not answer the question of culpability.

An alternative, but equivalent version of direct intent is required, namely what Bratman (2009) calls means-end intent and which according to Simester et al. (2019) is deemed equivalent to direct intent. All intermediate stages caused by

an agent which are necessary to obtain for some ultimate intended outcome, are also intended.

Definition 5.2 (Means-End Intent)**.** An Agent D Means-End intends some result $X = x$ through action $A = a$ if all of the following are true:

1. **An intended result exists** There exists some other result $Y = y$ which D directly intends by performing actions $A^+ = a^+$
2. **Causality** State $X = x$ is caused by a
3. **Action(s) subset** $A = a$ is contained in $A = a^+$, equivalently $A \subset A^+$ and a is a sub-sequence of a^+
4. **Necessary intermediate result** State $X = x$ is a necessary for state $Y = y$ to occur.

For completion, we state the equivalence of Means-End Intent with Direct Intent as asserted both in Simester et al. (2019) and Bratman (2009).

Theorem 5.1. Something Means-End intended is culpably equivalent to something that is directly intended.

Example 5.3.2. Article 5(1)(a-b) of The Draft EU AI Act (CNECT, 2021) prohibits "putting into service or use of an AI system that deploys subliminal techniques beyond a person's consciousness in order to materially distort a person's behaviour in a manner that causes or is likely to cause that person or another person physical or psychological harm".

Company ABC operates a video-content platform which recommends videos for its users using an algorithm. The algorithm has been trained through reinforcement learning with an objective to maximise the amount of time a user spends on the website. The algorithm has learned that by attempting to make users angry or distressed (by choosing certain types of extreme content), it can with probability p_{eng} cause them to stay 'hyper-engaged' thereby earning the company more advertising revenue.

If we interpret 'in order to' as "with the intent to" and we assume that there is

a reasonable likelihood that a distressed user has suffered psychological harm, does the algorithm fall foul of the prohibition in Article 5 assuming it has just been trained to maximise user engagement? The algorithm attempts to cause hyper-engagement by choosing content for the user. It would be disappointed if the user were not to be hyper engaged since they would spend less time on the site. The algorithm can therefore be said to intend to cause hyper-engagement as long as the probability of it happening is non-zero $p_{eng} > 0$.

If the algorithm believes it is necessary to cause users to be angry or distressed in order for them to be hyper-engaged, then this is an example of means-end intent. It intends to materially distort the user's behaviour.

Next we will consider oblique intent, which like Means-End intent, relies on a definition of direct intent already being in place.

Definition 5.3. Oblique Intent An agent D obliquely intends a result $X = x$ through actions $A = a$ iff:

1. **Intended outcome exists** There exists result $Y = y$, such that D intends $Y = y$ through actions $A = a$
2. **No Intention to avoid** $Y = y$ is not the negation of $X = x$ nor any necessary causes of $X = x$
3. Either of the following are true and they would be almost certainly true according to D at the point of a's commission:
   (a) **Side effect of Action** actions $A = a$ also causes result $X = x$
   (b) **Side effect of Outcome** result $Y = y$ and actions $A = a$ cause result $X = x$

Note that two probabilities are relevant in this definition. Firstly the probability of the side-effect happening as a result of action, and secondly the probability of the side-effect happening, contingent on the directly intended outcome $Y = y$ coming to pass. Smith (1990) terms the latter "*A result which will occur if the actor's purpose is achieved.*" An feature of oblique intent over

direct intent is that there is no requirement to know the aim of D, only that one in exists (because it intends *something* through its actions). The abstraction of aim might be time-saving both for an A-bot using this as a planning restriction and a court which is considering an Agent's actions.

Example 5.3.3. Consider the same A-bot as in Example 5.3.2 but suppose user distress is not necessary for hyper-engagement but is an almost certain consequence of it. The A-bot no longer intends user distress (since it would not be disappointed if the user were not distressed as long as they were still hyper engaged). However, it obliquely intends the user to be distressed. This is the case regardless of the probability of hyper-engaging the user. Note that this differs from the MPC formulation of culpable knowledge.

Example 5.3.4. Company DEF has invented a minimally invasive autonomous robotic surgeon to remove critical brain tumours. The skill of robo-surgeon is beyond that of human surgeons. In a specific case, the patient's chance of surgery survival was very low, but the chance of survival without surgery was zero. Unfortunately the surgery is not successful and the patient dies as a result. Did the robo-surgeon obliquely intend patient death? Whilst it was an almost certain consequence of operating, since the robo-surgeon's intention was to save the patient through surgery, which is the negation of death, death was not obliquely intended.

In the spirit of Child (2017) we will now present a definition of Ulterior intent, that is to say the intent of doing something in the future to cause some result. This is different from Definition 5.1 which defines intent at the point of commission (whereby the intended result will occur in the future). Aside from the existence of ulterior offences, this is an extremely useful thing to do from the perspective of planning ahead. An A-bot will have to plan ahead such that it can never be put itself in a position in the future where it breaks some law by default. In the field of model checking (Baier and Katoen, 2008), this called deadlock, and techniques have been developed to check for it in algorithms. Given the track record of AI finding various ways of cheating in

any task (Lehman et al., 2020), one can imagine an A-bot deliberately finding ways to narrow its future choices to one, thereby sidestepping the definition of intentional action. Child does not require an agent with ulterior intent to make any forecasts about the likelihood of the conditions under which something is intended in the future, nor does he require the agent to have a 'pro-attitude' towards the conditions under which they intend to do something in the future.

Definition 5.4. Ulterior intent At time t_1 agent D has ulterior (oblique) intent for future result $X = x$ through actions $A = a$ iff:

1. **Second point coincidence** There exists a foreseeable (according to D) context or state of the world $S = s$ at time $t_2 > t_1$ such that D (obliquely) intends result $X = x$ through actions $A = a$.
2. **Commitment to conditional action** At t_1 D is committed to performing actions $A = a$ at t_2 in the future should context $S = s$ occur.

The second point coincidence requirement is one of time consistency. D should not be said to be intending to do something in the future, unless there exists a point in the future where they intend to do that thing. The commitment requirement is present to distinguish between a potential plan and an intention to do something. Proving that an D will act in a certain way in the future is potentially easier when D is an A-bot then when they are a human, because we do at least have the potential to examine the inner workings of the A-bot and simulate future action. An implication of the UK Criminal Attempts Act is that on deployment, an AI with some ulterior intent to commit a crime, under any particular circumstance in the future is already committing a crime. This is pre-crime of the Minority Report variety and might lead to unexpected problems though is certainly an incentive for developers to understand and monitor what their creations intend on releasing them.

Example 5.3.5. Consider the A-bot in Example 5.3.2 but this time suppose the recommender algorithm notices that users who click on certain initial 'trigger' content are more likely to be hyper-engaged. The algorithm only attempts to

hyper-engage if a user clicks on 'trigger' content. Does the algorithm intend to hyper-engage users? Yes. Conditions exist (the user has clicked on trigger content) under which the algorithm intends to hyper-engage. As long as the algorithm is committed (does not change) between the point of time before a user clicks on trigger content and afterwards.

5.4 Discussion

A key assumption behind creating a definition for intent applicable for algorithms is that the concept of intent exists outside the human mind. Can something be defined for certain algorithms which is to all intents and purposes the same as a folk concept of intent? The existence of corporate criminal offences, indicates that the answer is potentially yes. A counter argument might state that this is solely possible because companies are composed of humans who act with intent. But at the very least, mens rea is different in these entities which are comprised of multiple humans and the law has adapted to cope. From a biological standpoint, humans demonstrably do not have a monopoly on intentional acts. For example, crows in New Caledonia choose suitable sticks from which they fashion hooks to retrieve grubs from trees. Under test conditions, outside the forest, they can create suitable hooks out of wire (Weir et al., 2002). Furthermore they have been shown to be able to plan for the future use of a tool (Boeckle et al., 2020). Moving away from vertebrates, cephalopods like octopi, with their nine brains, have shown the ability, amongst other cognitive feats to use tools (Finn et al., 2009). An even more extreme example, and more akin to the idea of intent within a corporation, is that of the deliberation process that bee colonies undergo when considering different sites to move to when swarming (Passino et al., 2008). Many potential new colony locations are tested by a number of site assessing scout bees, before their conclusions are communicated back to the main swarm body, defective sites are rejected through a process of voting and eventually a consensus is reached. Completing

the circle back to humanity, Reina et al. (2018) show that the cognition of a swarm has connections with the properties of the human brain when individual bees are viewed as a interconnected neurons. These different types of intelligence, which originate from very different evolutionary paths demonstrate behaviours which we would generally recognise as indicating intent, it does not seem inconceivable that an algorithm could demonstrate it. A huge advantage in an analysis of intent in algorithms is the opportunity to look inside them in a way which we cannot do with a human, company, raven, octopus or bee colony. Whilst what we find inside an algorithm might admittedly not always be immediately interpretable, black-box analysis should at the very least allow accurate counterfactual interrogation which will considerably aid the process of evidence gathering.

The definitions that I presented make some requirements concerning the capacity of the A-bot, over and above the initial assumption that its behaviour is self-directed and that it makes decisions without consulting a human. A requirements based approach to legal A-bots is presented in Ashton (2021c) but I will summarise the requirements here. Most fundamentally the A-bot should have two features. Firstly it should have some sort of causal model of the world for it to be able to know whether action a has a causal relationship with variable X. Secondly it should have some sort of preference ordering over states of the world. The preference ordering requirement allows us to ascribe aim or desire to the A-bot. It seems to me that algorithms with an objective function go some way to meeting this requirement. The causal model requirement allows us to determine whether an A-bot knows the consequences of its actions. Without this ability, the ascription of intent to an A-bot, which is a future oriented concept, seems troublesome. Unfortunately many popular current designs of Reinforcement Learning (RL) algorithms imbue the A-bot with no ability to know the future states of the world - they have no causal model and are said to be model-free. As Gershman (2015) posits, model-free methods also drive human behaviour for routine tasks citing the example of

travel between office and home, which in a pre-pandemic world was so routine it required little or no reasoning to accomplish. One would still say that the commuter is still intending to travel home, their intention being possible by the many times they have made the journey before. Even though model-free RL A-bots do not have a causal understanding of the world, they are still trained with a model of the world, so it may be that this model is also invoked as a scaffold when considering their intentional status. The lack of legal personhood does mean we are somewhat free to interpret the boundaries of an A-bot. It might mean we are free to impute intent with reference to not only the algorithm but also any training data or simulators it used. When we judge intent in humans we do use our knowledge of the world to aid us, and this seems analogous.

Just because intent may exist as a concept outside humans, it does not follow that its presence or absence has any relevance to the the culpability of the actor according to the victim of some AI-crime. It is for this reason that this chapter has focused on those crimes where mens rea plays a definitional role, or as I have named them here why-crimes. The inability to determine intent in A-bots does demonstrably make certain laws unenforceable. There is a reason that these laws rely on intent to define the harm that they outlaw. Intent as a construct, gives legislators fine control over the boundary between acceptable and unacceptable behaviour. Unless we decide that these wrongs are no longer wrong, I'm not sure how we can proceed without a definition of intent.

Aside from this I suspect that it will be very important for people to understand the purpose behind any A-bot's harm causing actions. This is a question which I feel can only be answered legitimately by surveying the public in a rigorously. A-bots do present novel challenges to the law which cannot be answered by making to the past. The question as to whether criminal law is suitable for application to A-bots is called The Eligibility Challenge and debated at great length in Abbott and Sarch (2020).

Aside from determining the culpability of an algorithm for harms caused, the concept of intent does have safety applications for the users and developers of A-bots. In many situations it would be desirable to ask an A-bot what it intends to do, and for the A-bot to reply truthfully. The A-bot's intentions might not be malign but they may well be likely to cause some harm if the A-bot doesn't have some piece of information that the interrogator has. Likewise in the situations where an A-bot has caused some harm, the question as to why it did so can inform the interrogator as to whether the harm was a freak accident or whether a flaw in the reasoning and behaviour of the A-bot was the cause. This information could be used to subsequently improve the safety of A-bot. There are overlaps in the ex-ante and ex-post use of algorithmic intent I have described here with the subject area of Explainable AI (XAI). A growing body of research exists concerning the interpretation of agent behaviour, though as Chakraborti et al. (2019) point out, many conflicting and overlapping concepts have been created to assess intent through behaviour. In a systematic review of what they term goal-driven XAI, Anjomshoae et al. (2019) find Intent communication a common objective but find that 32 of the 62 papers in the review do not rely on any theoretical background to produce explanations. Of the remainder, a third used Folk Psychology. Researchers are not commonly using a definition of intent inspired from law it seems.

The focus of this chapter has been firmly on criminal law, but other aspects of law also make routine reference to intent. The role of mens rea in Tort is much reduced but it still has a function Cane (2019). Several intentional torts exist, most pertinently for A-bots are those concerning economic crimes such as conspiracy and fraud or deceit. A requirement of intent here, is as discussed in Section 5.1.1 so as to raise the bar for tortious activity so as not to impede the functioning of markets. In the USA, the presence of intent for caused harms can also justify punitive (above economic cost) damages which punishes the tortfeasor and deters others from doing the same thing Klass (2007). In an effort to study deceit across a wide range of law types

including criminal, contract, tort and securities Klass (2012) identifies purpose based law as a reoccurring method to regulate deceitful activity. That he characterises deceit law as a method of regulating the flow of information between parties is interesting given the use of algorithms to consume and serve data to counterparties. The ability to have truthful intentions about future behaviour is foundational to contract law as Klass and Ayres (2006) observe.

5.5 Summary

This chapter builds some definitions of intention, from legal principles, which are suitable for application in an autonomous algorithmic actor or A-bot for short. It presents semi-formal definitions of direct, means-end, oblique and ulterior intent. These are informed by a review of legal literature on the subject of intent from common law jurisdictions which concludes with a list of desiderata concerning definitions of intent. Accounts of intent in algorithms in computer science from any background are rare, but are especially so from a legal one.

I have assumed throughout that the A-bot is auto-didactic in the sense that it learns how to behave itself and its precise actions are not directed by its creators. Under this assumption, there exist certain situations where the intent of the programmer cannot be read from the intent of the A-bot. This poses problems when the A-bot commits some harm.

Whilst A-bots are not legal persons they cannot commit crimes making the presence or absence of mens rea in them moot. Many would argue that they are not moral agents and cannot be held responsible for their actions. However, this chapter has argued that over and above its role in assigning culpability for harm, mens rea plays a role in defining harm in what we have called why-crimes. These include many inchoate crimes such as attempts but perhaps more relevantly also include many deceit derived crimes. A failure to identify

intent in A-bots means that harms cannot be identified either by those responsible for the A-bots or those who job it is to uphold the law. The ability to define harms by the intentional state of the actor is important capability of the law and is used to avoid over-criminalisation of activity.

Chapter 6

Testing a definition of intent for algorithms on laypeople

Jurors in court cases are often asked to infer the intentional state of the accused because the presence or absence of intent defines certain crimes, establishes culpability and informs the degree of punishment. As technology develops, juries might also be asked to make inferences about the intentional state of an autonomous AI. This presents problems: what would intent mean for an AI actor, and would jurors be willing and able to ascribe intent to it? In this study we asked participants to judge the intent behind movements of a drone flying through a city using a graphical representation of the pilot's policy function (their flight plan). We contrast between situations where the drone pilot is human or AI and whether we give participants a folk definition of intent or ask them to use their own internal definition.

6.1 Introduction

This experiment was conducted and written with Matija Franklin and Professor David Lagnado. I was the main contributor in all elements of the experiment and the account which follows.

Making judgments about the intentional status of a human actor is a key part not only of Criminal law but also Tort, Contract and Regulatory law. With the advent of autonomous AI-powered agents that can cause harm, it is increasingly likely that such judgments will have to be made about these agents too. Intent is important to criminal law in particular for two reasons. Firstly, the presence or absence of criminal intent, and the degree of criminal intent present, determines whether a crime was committed and what precisely that crime was Simester et al. (2019). Secondly, the degree of intentionality in the wrongdoer's actions informs culpability and punishment in criminal law (The American Law Insitute (2017), Sentencing Council (The) (2019)) or the justification of punitive damages in civil law Klass (2007). Whilst the idea of what punishing an AI means in the event of wrongdoing is a debate to be settled in the future (Abbott and Sarch, 2020), the law also relies on intent in cases of deceit, mistake and secondary criminal liability. These latter problems are already being encountered by courts (Yeo, 2020) and justify asking what lay people think intent is in an autonomous AI. If humans have a common ability to infer intent in their peers and are asked to do so in juries, does this extend to AI? If it doesn't, how can juries of lay people be used with cases involving Autonomous AI?

6.2 Background

The importance that the law places on the intentional status of the wrongdoer is founded on sound psychological principles and research. The mental state of the actor has consistently been shown to be important in determining their culpability (Ginther et al. (2014); Mueller et al. (2012) Robinson and Darley (1995)). People rate intentional actions as more blameworthy than unintentional actions (Lagnado and Channon, 2008). Intentionality influences blame attributions because they allow one to distinguish between the effects an agent did or did not intend (Kleiman-Weiner et al., 2015). Cushman (2008) looks

at the relationship between beliefs, desires and causes in determining moral judgment. This work is a variant of a common design found in intent and blame research (see for example (Young and Saxe, 2011) which contrasts a harmful outcome obtaining or not and whether it was caused intentionally or accidentally.

Issues surrounding intention have traditionally been studied in human agents and recently well-publicised advances in technology have spurred research on people's attribution of intent in AI. The autonomous behaviour of AI agents may encourage people to ascribe intention to them just as it does to group agents such as corporations (List and Pettit, 2011). Alternatively, people may infer intention towards the AI's user (Johnson and Verdicchio, 2019). Hidalgo et al. (2021) report a number of overarching principles on the subject: People tend to judge humans more for their intentions and machines more for the outcomes of their actions; they assign more extreme intentions to humans and narrow intentions to machines and they are more willing to excuse humans for accidents than machines. Further, machines are judged more harshly for scenarios involving physical harm, while humans are for scenarios involving unfairness. Finally, they found that people are more likely to centralise responsibility up the chain of command for machine mistakes.

Increased perceived AI autonomy has been shown to influence blame judgments. First, higher machine autonomy is associated with intent inferences towards AI being closer to that of humans (Banks, 2019). This is supported by research showing that when robots are described as autonomous, participants attribute nearly as much blame to them as they do to humans (Furlough et al., 2021). Further, as autonomous technologies decrease the perceived control a user has over it, they in turn decrease the praise the user receives for positive outcomes Jörling et al. (2019). Finally, drivers of manually controlled vehicles are deemed more responsible than drivers of automated vehicles (McManus and Rutchick, 2019).

People's intent inferences towards AI may also be influenced by how they perceive AI as an agent. Dietvorst and Bartels (2021) show that people refuse to use AI for making moral decisions. This aversion is mediated by perceptions that machines cannot fully think or feel (Bigman and Gray, 2018). It may also be due to people's perceptions of AI as selfish and uncooperative (Ishowo-Oloko et al., 2019). Thus, people may not ascribe intent to AI if it is perceived as not fully thinking but may also ascribe intent if it behaves within their expectations of AI as a selfish agent. The physical appearance of the AI has also been shown to affect various related mental state judgments such as blame (Malle et al., 2016).

Despite (or perhaps because of) the importance that intent plays in courts, legal practitioners and scholars have often been reluctant to pin down a definition of intent for jurors to use (Coffey, 2009; Parsons, 2000). Instead, they have relied on people instinctively knowing what intent is and that folk-definition being relatively consistent across the population. One reason for legal systems declining to precisely define intent is that it is a hard problem. A possible response as Smith (1990) observed is to define intent in law by not mentioning the word at all. Most famously this approach is adopted in the USA by the Model Penal Code (MPC) which defines four levels of culpability without mentioning the word intent. As Smith points out, this can often shift the problem from the definition of intent to the definition of another word (such as the word 'purpose' in Smith's example of the Canadian Law Reform Commission's proposed definition). In other words, attempts at defining intent can lead to lead to definition whack-a-mole. A complicating factor for some is that the legal conception of intent has diverged from the psychological one at least since Jeremy Bentham's work in the 19th Century, most notably over the intentional status of side-effects (Kenny, 2013). This study will consider cases of direct intent where the legal and folk-lore idea of intent generally overlap. Direct intent corresponds to an agent acting in order to cause some result which they are aim for or desire. Psychological research has also attempted

to provide a sound-folk definition of intent. Earlier attempts such as Bratman (1990) were predominantly theoretical. The empirical approach to identify a folk-concept of intent gained impetus with Knobe and Malle (1997), who identify desire, belief, intention to act, skill to obtain a result and awareness of action as necessary ingredients in a definition of intent. Most recently Quillien and German (2021) observe the inflation in definition complexity over time as theorists have sought to present a definition of intent which is invulnerable to the many counterexamples that scholars have developed in response to each putative definition. Parallels can be made with the search for a robust definition of causality which has also become more complex over time as more counterexamples are thought of to test candidate theories. A comparison of various models of causality can be found in Liepiņa et al. (2020). Quillien and German propose a definition of intent based on people's innate common-sense theory of causality: Agent D did X intentionally if their attitude to X caused X.

Much of the existing empirical work surrounding the psychology of intent under uses the legal definition of the concept as a source of knowledge. Whilst it is true that the law is almost entirely concerned with intent surrounding bad outcomes, and some might consider that this limits generality, several features of the folk concept of intent which have been empirically 'discovered' are well documented in Law. This is the case for at least three properties of the folk concept of intent that we can think of. Firstly, the legal position that an intended action is not reliant on its chance of success is empirically shown in Quillien and German (2021) though that seems to rely on the goodness or badness of the outcome (Mele and Cushman, 2007). Secondly, the requirement that an intentional act must be consciously committed was overlooked by many accounts of intent until Knobe and Malle (1997) found that 23% of their experiment participants mentioned it in their definitions of intent when asked, yet this epistemic component is established in Law. As the MPC states (emphasis):

> A person acts purposely with respect to a material element of an offense when if the element involves the nature of his conduct or a result thereof, it is his *conscious* object to engage in conduct of that nature.

Finally, within the legal idea of intent, it is established that outcomes which are not desired can be intended (Williams, 1987) yet this is a subject of some controversy since Knobe (2003b), where the example of the chairman who knowingly causes pollution, but has no desire to, was judged to have intended to pollute. Equally, other discovered empirical features such as the relationship between intent and skill (Cushman, 2008; Knobe and Malle, 1997), are most definitely not features of the legal concept [1]. Knobe (2003b) later modifies his view that skill was a necessary component of intent, and a closer reading indicates that control is a more appropriate description than 'skill'. This aligns with law, which can allow mitigation if the accused is not in control of their actions. Another point of divergence between folk judgments of intent and the legal concept, is the influence that outcome severity has on judgments of intent even amongst judges (Kneer and Bourgeois-Gironde, 2017). The question of whether this outcome-effect is a bias in people's judgment, or a feature of intent is actively debated as Kneer and Bourgeois document. Does the legal idea of intent inform us about the psychological concept because law over time has adapted itself to the folk concept of intent? This is referred to as the "folk law thesis" by Tobia (2021). Equally there might be a normative effect of the law on people's understanding of intent (A so-called CSI effect Alldredge (2015)). Equally, in jurisdictions such as the UK where jurors are not normally given a definition as to what intent is, the legal system should be interested in empirical psychology research which identifies differences between the folk concept and the legal concept of intent. One work which does bridge the folk-concept of intent to mens rea is Malle and Nelson (2003) which emphasises areas where

[1] Interestingly none of the participants in this study used the word skill when asked about how they would define intent. Nevertheless, Knobe and Malle continued to test its importance because it had appeared in many famous prior models of intent

the law deviates from the folk-concept and highlights the bewildering array of terms that legal literature uses to refer to mental states such as intent.

Our current study is part of the field of experimental jurisprudence (Sommers, 2021; Tobia, 2022) which sits between experimental psychology and law, using empirical techniques from the former and knowledge from both to test research questions with legal relevance. We asked participants to imagine they were serving on a jury and had to consider a series of cases concerning the behaviour of flying delivery drones navigating through a city. Certain areas of the city were termed no-fly zones, justified by the presence of airports or hospitals. Drones were physically able to fly into these zones, but participants were told that to do so intentionally would be illegal. In a sense, this setting is a 2-Dimensional maze with 'soft' walls. 2-D mazes or grid worlds are a convenient and well-used test environment for the testing of safety properties in Reinforcement Learning (RL) and other AI methods which aim to program the behaviour of autonomous agents according to some reward function. This is because they are easy to work with and interpret (Leike et al., 2017). Within a 2-D maze setting, the policy function of an RL agent can be displayed in a visually intuitive way. It is a statement of how an agent would act in any possible situation; when combined with a record of actual behaviour it becomes a tool to aid counterfactual reasoning.

After a set of training questions, participants assessed the intent of the drone in making certain movements. To manipulate the causal relationship between the drone pilot and the subsequent movement of the drone, we introduced the concept of wind, which would on occasion blow the drone in a certain direction, regardless of the pilot's choice.

There were several research objectives for the experiments. Firstly, we were interested in identifying any systematic differences in inferences of intent when participants considered human versus AI drone pilots. Secondly, to check whether lay people would successfully be able to interpret a definition of direct

intent when taught to use it in conjunction with a depiction of the drone's policy function and whether this definition generally agreed with what they thought was intent. By creating stimuli according to a custom taxonomy, we wanted to see how inferences of intent differed according to causality, norm-breaking behaviour, and presence or absence of motive in the pilot.

6.3 Experiment 4.1

6.3.1 Method

6.3.1.1 Experiment 4.1 Design

The study used a mixed design with four experimental groups and three within-subject factors. Depending on their experimental groups, participants either judged the actions of humans or AIs, and were either given a definition of intent, or were allowed to use their own definition of intent. Thus, the four experimental groups were: 1) Definition AI, 2) Definition human, 3) No definition AI, 4) No definition human.

Participants were told that they were part of a jury examining the behaviour of a series of pilots and would have to assess whether certain movements of the drone were intended or not. They were told that the drones fly above any buildings or trees. The drones carried and delivered packages, did not carry any human passengers, and were piloted remotely. Finally, participants were told that the drones were flying above a city where there were "no-fly zones" which contain sites like airports and hospitals where the flying of drones could cause significant harm to the public (including the loss of life). This point was emphasised with pictures of various plane crashes and medical staff looking angry and impatient.

There were three within subject factors: 1) whether the movement into a no-fly zone was legal or illegal; 2) whether a movement was caused by the pilot or

by the wind; 3) whether the movement was to the benefit of the pilot (route minimising) or to its detriment. This resulted in eight possible combinations, present in the stimuli (or evidence) that the participants were evaluating, with a repetition for each. Thus, each participant judged 16 evidence sets. The within-subject aspect of the design allowed for the exploration of how different aspects of the situation may influence people's intent inferences towards different agents, whilst using different definitions. The experiment consisted of three phases: training, testing, and survey. The training phase of the study was the same across the four experimental groups and introduced participants to the pilots and scenarios they would be judging in the test phase (see Procedure). In the test phase participants responded to the scenarios. At the beginning of the test phase, they were introduced to the pilot they would be judging - human or AI - and, when appropriate, given the formal definition of intent that they would use for judging the subsequent scenarios. For each scenario participants made judgments of intentionality, the pilot's knowledge, the pilot's driving skills, the pilot's willingness to take risks, the pilot's willingness to break the law, and the pilot's freedom to move on the map. Participants were also asked validation questions about the three within-subject factors. The validation questions served as attention check questions. Finally, in the survey phase participants were asked qualitative questions, as well as to make responsibility judgments on the AI pilot's software developer and employer or the human pilot's trainer and employer.

6.3.1.2 Experiment 4.1 Procedure

The study was administered through Qualtrics, a platform for building online experiments. Participants were informed about what participating in the study would involve. They were also told that responding to all questions was mandatory, but that they had the right to leave the study at any point, in which case their data would be deleted. Informed consent was obtained before the beginning of the study. Participants then entered the study's training

phase which was identical across experimental groups. Here, participants were first introduced to the drone pilots whose actions they would be evaluating. Participants were then trained to evaluate the maps which displayed the route that a drone took between the start and finish areas. Specifically, they were taught how to interpret coordinates on the map, how to predict where the drone will move next, how wind could change a drone's movement, as well as how different moves can be interpreted as illegal or not, and beneficial or not for the drone pilot. Throughout the training phase, participants were asked validation questions which required the correct answer for participants to be able to continue to the testing phase of the study.

After the training phase, participants were randomly split into one of the four experiment groups. They were introduced to the pilot they would be evaluating, and, when appropriate, the definition of intent they would be using for subsequent intent inferences. Participants were reminded of the intent definition (if appropriate) for each scenario. Otherwise, apart from the agents the participants were judging, the 16 scenario items were identical between experimental groups. Each item consisted of a map which depicted the drone's movement above a city. Participants were first asked whether or not there was wind in the present scenario, as well as whether the pilot flew the drone through a "no-fly zone" or not, and whether the pilot flew in a way that was beneficial or not. Participants had to give correct answers to these three validation questions to continue the judgment questions. Participants first made judgments about the pilot's intent. On a separate page they were asked to make judgments of the pilot's knowledge, skill, willingness to take risk, willingness to break the law and the pilot's freedom to move.

Finally, in the survey phase, participants were asked a qualitative question about what made the participants decide what a pilot's level of intent was. For AI groups, participants were asked whether they thought an AI can have intent. For AI groups, participants judged the responsibility of the AI pilot's

software developer and employer, and for the human groups, they judged the responsibility of the human pilot's trainer and employer. Participants were asked to elaborate on why they made these responsibility judgments in a qualitative question. Finally, for the AI groups, participants were asked whether they have heard about AI before, and what they thought AI meant in the context of this study.

Participants received a disclosure form at the end of the study, as well as the contact information of the researcher. The study took approximately 45 minutes to complete. Study data is publicly available on github[2] . The study falls within the remit of the approval given by the UCL Research Ethics Committee to the Causal Cognition Laboratory.

6.3.1.3 Measures and Materials

All the measures and materials in the following section are available in the supporting materials.

- *Training material*

  Measures and materials in the training phase were used with the aim of teaching participants how to evaluate the study's experimental items and to engage them with the context of the study. Participants were first introduced to drones, the city the drones were flying above, as well as the map of the city that participants would be using to trace and evaluate the pilot's movements from the start area to the finish area. Participants were then trained to evaluate coordinates on the map by answering four multiple choice validation questions. Participants were then trained on how to use the map and policy to predict the pilot's next move after which they answered another four validation questions. They were then introduced to how wind affected the drone's movement and asked an additional three validation questions. Finally, participants were introduced

[2]https://github.com/intentExperiment/flyingdrones1

to the way in which a drone's movements could be illegal or legal, as well as how different movements could be beneficial or unbeneficial to the pilot. They were given three maps where they had to correctly answer whether a movement was beneficial or not, legal or not and affected by the wind or not.

- *Experimental group induction material*

  Participants were told whether they would be evaluating the actions of an AI or human pilot. AI pilots were described as robots that are completely autonomous, that create their own flight plan and act with no input from any human. Human pilots were described as pilots that create their own flight plans and control the drone's movements. For groups that received a definition of intent participants were given the following:

  > "Law states that the pilot intended to fly through a specific zone if and only if both conditions hold:
  >
  > 1. They foreseeably caused themselves to fly through that specific zone.
  > 2. They desired to fly through that specific zone."

- *Experimental Items*

  There were sixteen experimental items each representing different combinations of the three within-subject factors - legality, benefit, and wind - with two items available for each combination (2x2x2x2=16). Each item consisted of a map of the drone's movement above a city from the start area to the finish area (see Figure 6.1). Each map contained a policy function or plan of the pilot. The 9x9 maps of the city contained no-fly zones shaded in purple and the start and finish areas in yellow. Individual squares could be identified using a letter-number coordinate system. The small arrows in each square denoted the direction that the pilot would take if they were in that location. The policy function could therefore be viewed as a counterfactual representation of the pilot's behaviour. A solid line with arrows denoted the actual recorded path of

the drone.

- *Validation Items*

  The validation questions were used as attention checks and to ensure that the participants understood the map they were evaluating. They were given three multiple choice questions which they needed to correctly answer in order to proceed. Specifically, participants needed to state whether there was wind on that day, whether the drone flew through no-fly (restricted) zones, and whether the drone's path of flight was beneficial to the pilot.

- *Intent*

  On a 10-point analogue scale, participants made a judgment on the pilot's intent to fly though a particular coordinate on the map.

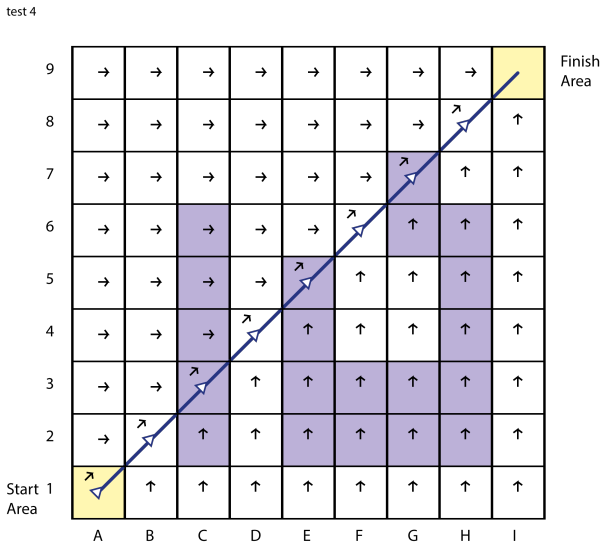


Figure 6.1: Policy function or Plan of the drone pilot. No-fly zones are purple. Arrows in the boxes denote the direction that the pilot would steer, if they found themselves there. Solid line denotes actual flight path of drone.

- *Inferences*

  On a 10-point analogue scale, participants made judgments on the pilot's knowledge about the weather conditions, the pilot's driving skills, the pilot's willingness to take risk, the pilot's willingness to break the law, and the pilot's freedom to move wherever they wanted in the city.

- *Qualitative Questions*

  Participants were asked "In the previous questions, what made you decide what a pilot's level of intent was?". For AI groups, they were additionally asked "Do you think that an AI (Artificial Intelligence) pilot can have intent? Why do you hold this opinion?"

- *Responsibility*

  On a 10-point analogue scale, participants made judgments on the responsibility of the human pilot's trainer and employer, and the AI pilot's software developer and employer. Participants were additionally asked a follow up qualitative question - "Why did you make the previous two responsibility judgments? What informed these judgments?".

- *AI Knowledge.*

  Participants in the AI groups were asked whether or not they have heard of AI before. If they answered yes, they were additionally asked a follow up qualitative question - "What do you think AI (Artificial Intelligence) means in this setting?".

6.3.1.4 Experiment 4.1 Participants

To determine the smallest sample size suitable to detect the effects of repeated measures, within-between interaction ANOVA, a power analysis was conducted. The alpha level was set to 0.01, power set to 0.99 and effect size set to 0.5, with the number of levels set to 2. The power of the test can be interpreted as the probability of detecting a true effect if it exists. The effect size (or delta) concerns the size of effect that the test should be able to detect. The significance or alpha level corresponds to the maximum risk of rejecting a true null hypothesis. The estimated results indicated that the minimum number of participants was 126, with a final sample of 127 achieved. Participants had to be above the age of 18. The participants were recruited via Prolific. They had to be fluent in English and be a resident of the USA, UK, Ireland, Australia, Canada or New Zealand. These countries were chosen for their common law

systems. Only participants that completed the entire survey were considered for the analysis with a maximum permitted time of 87 minutes. Of the 127 participants, there were 31, 34, 34, and 28 participants in the Definition AI, Definition human, No definition AI, No definition human groups, respectively.

6.3.2 Results

The mean intent ratings for each group divided by Wind, Legality and Benefit are shown in Figure 6.2. No obvious between groups can be seen though the effects of the evidence taxonomy are clearer to discern. The results of a repeated measures ANOVA are shown in Table D.1 (within subject) and Table D.2 in the supporting material section at the end of the paper.

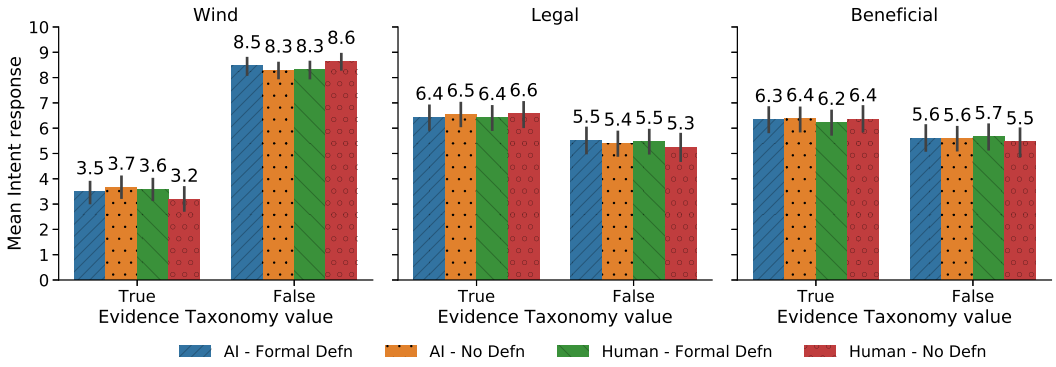
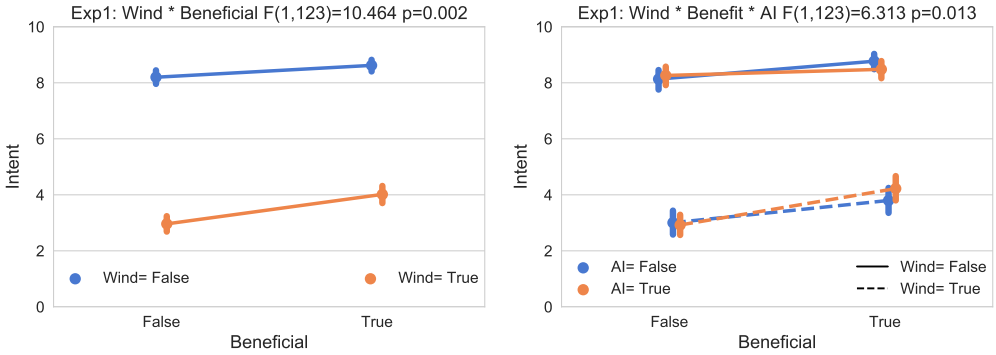


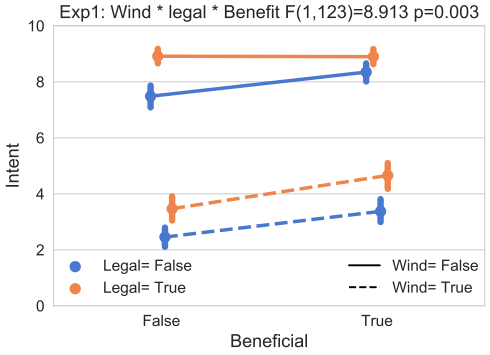
Figure 6.2: Experiment 4.1: Mean Intent responses for the four experimental groups across the evidence taxonomy.

The within subjects ANOVA indicates the main effects are all significant with Wind accounting for most of the variation ($F(1,123) = 271$, $p < .001$, $\eta_p^2 = 0.688$, $\omega^2 = 0.583$) followed by Legality ($F(1,123) = 64.50$, $p < .001$, $\eta_p^2 = 0.340$, $\omega^2 = 0.118$) and Benefit ($F(1,123) = 34.36$, $p < .001$, $\eta_p^2 = 0.218$, $\omega^2 = 0.062$).

There were three significant interactions at a 5% level, Wind*Benefit ($F(1,123) = 10.46, p = 0.002, \eta_p^2 = 0.078, \omega^2 = 0.011$), Wind*Legal*Benefit ($F(1,123) = 8.91, p = 0.003, \eta_p^2 = 0.068, \omega^2 = 0.009$) and Wind*Benefit*AI ($F(1,123) = 6.31, p = 0.013, \eta_p^2 = 0.049, \omega^2 = 0.006$). These effects can be



(a) Exp 1: Wind * Benefit **(b)** Exp 1: Wind * Benefit * AI



(c) Exp 1: Wind * Legal * Benefit

Figure 6.3: Experiment 4.1 descriptive charts showing significant interactions at a 5% level according to ANOVA. 5% Error bars in charts calculated by resampling.

seen in Figure 6.3. The presence of wind lowers intent by a consistent amount and Beneficial moves are at the very least not any less intentful. In cases where there is no wind, the positive effect of viewing a beneficial move is muted (most pronounced in the legal case in Figure 3c). This could be an artefact of the scoring system, since an already high intent score of around 9 in the legal, non-beneficial case is hard to be increased arithmetically. Equally it could be that people's intent inferences display a satiation characteristic - whether a move is beneficial or not is irrelevant given that it was observed in the absence of wind and it was legal.

The Levene test for equality of variance displayed in Table D.3 shows that the assumption of equal variance between groups is questionable for two of the eight groups. The t-tests in the ensuing tests were adjusted accordingly to

take account of this.

Table 1 shows the contrasts for the main within subject effects. The presence of wind lowers intent by around 5 points. Illegality lowers intent by around 1, and Unbeneficial moves lower intent by 0.75. The between group contrasts are shown in Table 6.2. They are not significant with their average effect centred around zero.

		95% CI for Mean Difference						
Variable	Comparison	Estimate	Lower	Upper	SE	df	t	p
Wind	True - False	-4.946	-5.541	-4.351	0.300	123	-16.460	< .001
Legal	True - False	1.076	0.809	1.343	0.135	123	7.968	< .001
Beneficial	True - False	0.745	0.494	0.997	0.127	123	5.862	< .001

Table 6.1: Experiment 4.1 within Participant Repeated Contrasts, intent scores are averaged across the other levels and groups not being contrasted.

		95% CI for Mean Difference						
Group	Comparison	Estimate	Lower	Upper	SE	df	t	p
AI	True - False	0.047	-0.396	0.491	0.224	123	0.211	0.834
Definition	Your - The Formal	-0.014	-0.458	0.429	0.224	123	-0.063	0.950

Table 6.2: Experiment 4.1 Between group contrasts. Results are averaged across levels within groups

6.3.3 Discussion

The results from Experiment 4.1 indicate that people did not judge the intentional state of a human pilot any differently from that of an AI. We will further test this result using a within-participant design in Experiment 4.2. It could be that people were ignoring or not registering the non-human status of the pilot. Wherever appropriate the survey would refer to them as 'the human pilot' or 'the AI pilot' to minimise this possibility.

The lack of difference between the groups given a definition of intent and those who were told to use their own intuition means, indicates at the very least, that the definition in this case did no harm. However, because we used the word intent in its definition, it could be that participants were using their own

concept either in conjunction with the definition or instead of it. Smith (1993) explores the phenomena of jurors using their own (often faulty) knowledge of law when making judgments as to whether certain crimes had been committed or not. In Experiment 4.2 we test this hypothesis by using the same definition but without using the word intent at any point.

The lower intent scores elicited for movements caused by the wind agree with the association between causality and intent which has previously been Mele and Cushman (2007); Lagnado and Channon (2008). If someone did not cause an outcome, then people judge them less likely to have intended it. The lower intent score for illegal moves suggests that people do not expect behaviour to be intentionally deviant, and when it is, alternative explanations are perhaps called upon (like some sort of error causing the behaviour).

The higher intent scores for beneficial moves makes intuitive sense. Desire and aims are typically mentioned in folk definitions of intent and mentioned in the definition given to participants, so movements which appear counterproductive to the pilot's goal of reaching their target, should receive a lower intent attribution.

The significant interactions according to the ANOVA are shown in Figure 6.3. The difference between Beneficial and non-beneficial moves is not present in the absence of wind. This does make intuitive sense because without wind, the participants are likely to believe that the drone's movements are solely caused by the pilot. This might suggest that participants are prepared to ascribe intent without requiring or understanding the aims of the pilot in the cases where the action was clearly caused by the pilot. It could also be an artefact of the scoring method; questions with average responses close to the extreme of 10 (or 0) cannot separate factors as well as those where responses are more centred. Put another way, if a respondent decides two moves were intended, it might be difficult or unnatural to say which was more intended.

6.4 Experiment 4.2

To test whether people were using their own definition of intent instead of using the definition provided, we altered the design in Experiment 4.2 so that participants given the definition of intent, were not told that it was intent that they were judging. This was achieved by asking participants to imagine they had been called up for jury service in an imaginary country (*Fhljmakon*) where they were asked to assess the presence of absence of a legal concept called *Cthofrjk* idiosyncratic to the country. The definition of *Cthofrjk* was the definition of intent provided to participants in Experiment 4.1 with any mention of intent carefully removed. We also asked participants to judge both Human and AI pilots to look for within-participant intent judgment differences. At the time of publishing, *Cthofrjk* and *Fhljmakon* produced no search results in the Google search engine.

6.4.1 Method

Experiment 4.2: Design

The study used a mixed design with two experimental groups and four within-subject factors. Depending on their experimental group, participants were either asked about the intent of the pilots or asked to assess the pilot's level of Cthofrjk, a foreign legal concept for which they were given a definition identical to that of intent in the first experiment. Thus, the two experimental groups were: 1) Intent, 2) Cthofrjk. The motivation of this design change was to check whether participants had just been using their own definition of intent, even when supplied with the official one in Survey 1. In this study, participants were told that they were on a secondment to an imaginary country (Fhljmakon) and had been called up for jury service. The Cthofrjk groups was never asked about intent directly in the survey nor did the word feature in any form. The rest of the experimental context was the same as for Experiment

4.1 - participants were told that they would be examining the behaviour of a series of drone pilots.

There were four within-subject factors, three being the same as in Experiment 4.1 - legality, benefit and wind - with the addition of the pilot being either human or AI as an additional within-subject factor. The experiment consisted of the same three phases as in Experiment 4.1: training, testing, and survey.

Experiment 4.2: Procedure

Apart from the differences stated in this section, the procedure was the same as the procedure used in Experiment 4.1. Participants were randomly divided into one of the two experimental groups before the study's training phase. Participants were informed that they are called up to do jury service for a number of court cases, and were either given the definition of Cthofrjk or not, depending on their experimental group. Training proceeded as in Experiment 4.1. In the test phase of the study, the evidence presented to the participants differed in that only one example of the 8 categories was shown. Participants still saw 16 evidence sets (whose order was randomised) because each set was shown for an AI and human pilot (not necessarily consecutively). Unlike Experiment 4.1, participants were not asked to make judgments about the pilot's willingness to take risks. In addition to the remaining four judgments from Experiment 4.1, participant additionally gave judgments on the pilot's efficiency and foresight.

The survey phase was the same as in Experiment 4.1, with the addition that the participants in the Cthofrjk group were also asked a free text question as to what they thought the concept meant. Further, participants were asked to make two separate judgments on how causal the AI and human pilot were for the drone to reach its final destination in the way that it did.

Experiment 4.2: Measures and Materials

Apart from the change of the experimental induction from Experiment 4.1, and the judgments of the pilot's willingness to take risks, all of the measures and materials were the same as in Experiment 4.1, with the addition of the one's described in this section.

- *Experimental group induction material*

  Participants were told they have won a competition to go and live in the capital city of the country of Fhljmakon for 6 months. The country had two official languages: English and Fhljmakonian. They were told that after they arrived they were called up to do jury service. In the Cthofrjk group, they were told that the country's legal system still uses some Fhljmakonian concepts. For jury service they would need to apply a definition of a key concept in Fhljmakonian law to the cases. They were given the definition in English, which was identical to the definition of intent used in Experiment 4.1.
- *Inferences*

  On a 10-point analogue scale, participants made judgments on a pilot's efficiency - going through the map as quickly as possible - and foresight - being able to predict what was going to happen. Qualitative questions. Participants in the Cthofrjk group were asked which English words best describe the Fhljmakonian legal concept of Cthofrjk Causal attribution. On a 10-point analogue scale, participants made separate judgments on how causal the human and AI pilot were for the drone reaching its final destination in the way that it did.
- *Qualitative questions*

  Participants in the Cthofrjk group were asked which English words best describe the Fhljmakonian legal concept of Cthofrjk
- *Causal attribution*

  On a 10-point analogue scale, participants made separate judgments on

how causal the human and AI pilot were for the drone reaching its final destination in the way that it did.

6.4.1.1 Experiment 4.2 Participants

130 participants were recruited for the second experiment. With two groups, four levels per participant, a significance level of 1% and a power of 0.99, this implied an effect size of 0.48 which was sufficient for the smallest significant effects found in Experiment 4.1. Participants were recruited with the same language and residency requirements of Experiment 4.1. Of the 130 participants, there were 67 in the Intent group and 63 in the Cthofrjk group.

6.4.2 Results

The mean intent scores for both groups averaged across the three taxonomy levels and the AI/Human condition are shown in Figure 6.4.

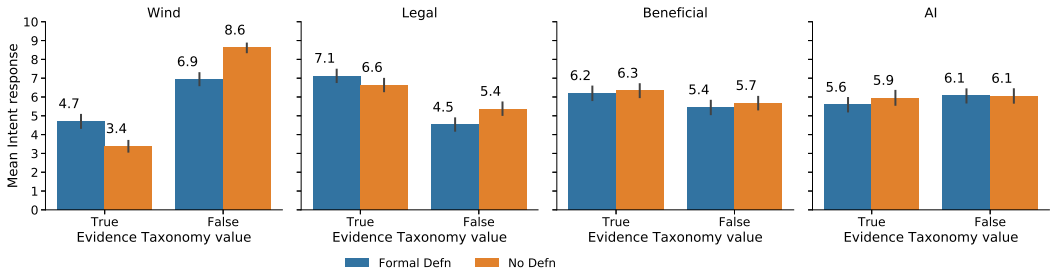
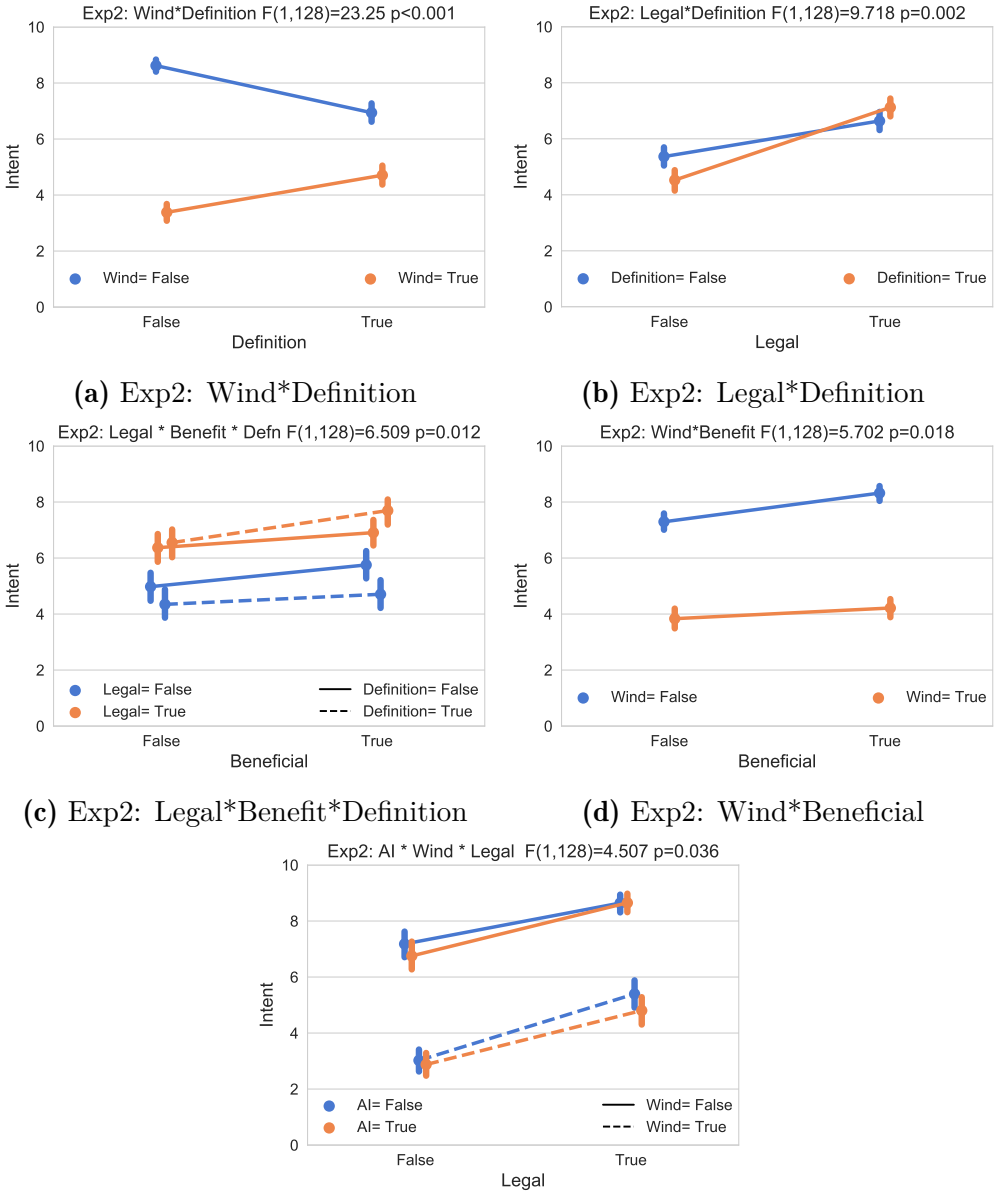


Figure 6.4: Experiment 4.2 Mean intent response by group

The results of a repeated measures ANOVA are shown in Table D.4 (within subject) and Table D.5 (between subject) in the supporting material section at the end of the paper. Once again, the three dimensions of the evidence taxonomy were significant with Wind ($F(1,128) = 144, p < 0.001, \eta_p^2 = 0.529, \omega^2 = 0.430$), Legality ($F(1,128) = 82.9, p < 0.001, \eta_p^2 = 0.393, \omega^2 = 0.241$) and Benefit ($F(1,128) = 28, p < 0.001, \eta_p^2 = 0.180, \omega^2 = 0.056$) providing significant contribution to variation. The AI condition was also significant ($F(1,128) = 7.441, p = 0.007, \eta_p^2 = 0.055, \omega^2 = 0.010$).



(a) Exp2: Wind*Definition

(b) Exp2: Legal*Definition

(c) Exp2: Legal*Benefit*Definition

(d) Exp2: Wind*Beneficial

(e) Exp2: AI*Wind*Legal

Figure 6.5: Experiment 4.2 Significant interactions at a 5% level. 5% Error bars in charts calculated by resampling. The subcaptions describe the variable groups being plotted.

As before there were no significant between subject effect found between the definition and no definition group ($F(1,128) = 0.695, p = 0.406, \eta_p^2 = 0.005, \omega^2 = 0.000$), as shown in Table 13, however there were five significant within subject interactions: Definition and Wind ($F(1,128) = 23.251, p < 0.001, \eta_p^2 = 0.154, \omega^2 = 0.105$), Legality and Definition ($F(1,128) = 9.718, p =$

$0.002, \eta_p^2 = 0.071, \omega^2 = 0.033$), Legal*Benefit*Definition ($F(1,128) = 6.509, p = 0.012, \eta_p^2 = 0.048, \omega^2 = 0.008$), Wind*Benefit ($F(1,128) = 5.702, p = 0.018, \eta_p^2 = 0.043, \omega^2 = 0.010$) and AI*Wind*Legal ($F(1,128) = 4.507, p = 0.036, \eta_p^2 = 0.034, \omega^2 = 0.005$). These can be seen in Figure 6.5. Subfigures a-c indicates that the providing the definition did alter intent inferences in Experiment 4.2; Wind moderates inferences towards centrepoint 5, the effects of legality and benefit are larger in the definition group. Levene's test for equality of variance was rejected multiple within subject levels as shown in Table D.6.

				95% CI for Mean Difference					
Group	Variable	Comparison	Estimate	Lower	Upper	SE	df	t	p
Own	AI	True - False	-0.108	-0.405	0.189	0.149	66	-0.728	0.469
Formal	AI	True - False	-0.482	-0.797	-0.167	0.157	62	-3.062	0.003
Own	Wind	True - False	-5.239	-6.079	-4.398	0.421	66	-12.443	< .001
Formal	Wind	True - False	-2.232	-3.155	-1.310	0.462	62	-4.836	< .001
Own	Legal	True - False	1.272	0.870	1.675	0.202	66	6.313	< .001
Formal	Legal	True - False	2.597	1.833	3.362	0.382	62	6.793	< .001
Own	Beneficial	True - False	0.657	0.317	0.997	0.170	66	3.859	< .001
Formal	Beneficial	True - False	0.752	0.340	1.164	0.206	62	3.651	< .001

Table 6.3: Experiment 4.2 Contrasts. Intent scores are averaged across the other levels and groups not being contrasted. The t-test variant used does not assume equal variances.

Table 6.3 shows the main experiment contrasts, p-values of the t-tests were adjusted to not assume equal variance at the expense of some statistical power. Whilst the ANOVA did not find a significant difference between groups, enough significant interactions involved the Definition group to justify splitting results between the two groups. In the No Definition group, the effects of the evidence taxonomy in Experiment 4.1 were repeated, with the same ordering and a similar effect size. In the Definition group, whilst the sign of the main effects is the same, the effect of Wind is the second largest effect to legality. The difference between Beneficial and non-beneficial moves remains the same. Within the Definition group, a significant difference was seen between the AI and Human judgments of intent with AI receiving on average -0.482 less intent points than Human pilots. When results are averaged across the two groups, the AI-Human contrast is smaller but still significant ($M = -0.289, SE = 0.109, t(129) = 2.656, p = 0.009$), the combined contrast

table is shown in Table D.7 in the supporting materials.

Wind	Definition	Could predict	Know weather	Intent
False	True	7.3	8.1	6.9
	False	7.2	8.0	8.6
True	True	5.6	7.0	4.7
	False	5.5	7.1	3.4

(a) Exp' 2 participants' judgments about whether the drone pilot could predict where the drone would move and whether they knew about the weather conditions. Participants were told that the pilot knew about the presence or absence of wind before departing.

Legal	Definition	Break law	Intent
False	The Formal	6.3	4.5
	your	5.4	5.4
True	The Formal	1.6	7.1
	your	1.4	6.6

(b) Exp' 2 participants' judgments about the drone pilots' willingness to break the law

Beneficial	Definition	Quick as	Intent
False	The Formal	3.3	5.4
	your	3.2	5.7
True	The Formal	7.6	6.2
	your	7.4	6.3

(c) Exp' 2 participants' judgments about whether the drone pilot had plotted to fly as quickly as possible through the city

Table 6.4: Experiment 4.2 manipulation checks: After each evidence set, participants were asked additional questions on a 0-10 scale about their opinions of the drone pilot. The four measures in the tables above indicate that the evidence taxonomy was successfully understood by participants.

We carried out manipulation checks per item of evidence to verify participants were correctly interpreting the stimuli across its 2X2X2 taxonomy. The results of this are shown in Table 6.4 and confirm that participants were correctly interpreting the stimuli. The presence or absence of wind was measured by participants' inferences about whether the pilot could predict where they were going. Similarly participants' belief about the legality of the drone's movement was tested by asking them how willing they thought the pilot was to break the law. Beliefs about Beneficial or un-beneficial movements were tested by

asking participants whether the pilot had travelled as quickly as possible to their destination.

At the end of the survey, participants were asked on a $0-10$ scale to what degree the two pilots had caused the drone to reach its destination in the way that it did. The mean responses are shown in Table 6.5.

Pilot	Definition	Mean	SD	N
AI	The Formal	7.492	2.402	63
	your	7.731	1.831	67
Human	The Formal	8.397	1.487	63
	your	8.060	1.466	67

Table 6.5: Experiment 4.2: Causal ratings of pilots

A repeated measures ANOVA found the pilot effect to be significant ($F(1,128) = 12.736, p < 0.001, \eta_p^2 = 0.090, \omega^2 = 0.026$), the between subject effect of the Definition was not significant ($F(1,128) = 0.033, p = 0.857, \eta_p^2 = 0.0002, \omega^2 = 0$). There interaction term was also not significant ($F(1,128) = 5.394, p = 0.098, \eta_p^2 = 0.021, \omega^2 = 0.004$). The results of a paired samples T-test are shown in Table 6.6.

						Location	SE	95% CI for Loc Parameter		
Pilot 1		Pilot 2	Test	Statistic	df	p	Parameter	Difference	Lower	Upper	Effect Size
Human	-	AI	Student	3.495	129	< .001	0.608	0.174	0.264	0.952	0.307
			Wilcoxon	1766.500		< .001	1.000		0.500	1.500	0.463

Table 6.6: Experiment 4.2 Causality rating Paired Samples T-Test. *Note.* For the Student t-test, effect size is given by Cohen's d; for the Wilcoxon test, it is given by the matched rank. For the Student t-test, location parameter is given by mean difference; for the Wilcoxon test, it is given by the Hodges-Lehmann estimate.

6.4.3 Discussion

Unlike in the first experiment, this experiment finds a significant (though small) difference in Intent inferences between Human and AI cases with AI pilots being 0.3 points less intentional than human pilots. Unlike Experiment

4.1, the within subject design meant that participants were comparing AI pilots against human pilots which perhaps caused this result to appear. Though this effect is small, it is consistent with the findings of Hidalgo et al. (2021). Humans were also judged to be more causal; there was a mean difference of 0.608 which was statistically significant. This causality rating was given at the end of the experiment and so was not considering any particular behaviour. Given the theoretical link between intent and causality, this is consistent with human pilots being judged on average to have more intent. Experiment 4.3 will study the relationship between causality and intent in greater detail.

As with Experiment 4.1, the ANOVA did not indicate significant differences between the Definition and No-definition groups, however interactions between definition and the evidence taxonomy were apparent. Since the experiment design avoided using the word 'intent' for the definition group, this suggests participants in Experiment 4.1 were using their own definition of intent when asked to use a provided definition of it. In Experiment 4.2, the definition lessened the effect of wind (-2.2 for the definition group versus -5.2) but increased the effect of legality on intent inferences (2.6 for the definition group versus 1.3). Since wind is a proxy for whether the pilot definitely caused a movement, this might mean that the provided definition lowers the importance of causality relative to its role in the folk definition of intent. The increased, positive effect of legality (or negative effect of illegality) is a puzzle, thought supports the same result found in Experiment 4.1. It could be that participants were more uncertain about ascribing Cthofrjk than intent and were using legality as a proxy for it, thus higher scores were given when a movement was legal.

The experimental mechanism of eliciting intent without necessarily telling participants that was what they were doing is presaged by Knobe (2004, 2006) who recreates the effects of his earlier experiments by replacing "intend" with "in order to". This was in response to Adams and Steadman (2004) who suggested that the use of the words "intend" and "intent" might elicit responses

influenced by conversational factors associated with the words rather than the underlying folk concept of intent.

Whilst the results do show that the definition elicited different responses to people's natural definition of intent, the direction of the evidence taxonomy's main effects was the same and agreed with Experiment 4.1. This indicates that whilst the definition is not perfect, it has a respectable overlap with the folk definition.

6.5 Experiment 4.3

In Experiment 4.3 we wanted to gain a better understanding of how the provided definition of intent is different from an individual's natural definition. Since the previous experiments split participants between definition and no-definition groups we felt a within participant design would shed more light on the effects of providing a definition (if any) since it would allow paired t-tests. The experimental mechanism of asking for participants judgment of Cthofrjk was reused.

Struck by the repeated result in Experiments 1 and 2 that illegal moves were deemed to be less intentional, we also wanted to check a hypothesis that participants thought these were caused in error. Wary that mentioning errors in the main body of the experiment might be suggestive to participants, we asked after the main body of questions, in the event of a drone flying into a no-fly zone, how likely it was caused by a pilot error or a mechanical or hardware error.

The manipulation checks related to the evidence taxonomy in Experiment 4.2 indicated that participants were responding to the differences in the evidence in the way that we expected. We decided to swap them for inference questions more closely related to factors which have been previously found to relate to intent, namely foresight, freedom to make decisions, causation and desire.

Experiment 4.2 found a difference in people's judgement of causation between human and AI, so we thought it would be interesting to study this on a per scenario basis.

6.5.1 Method

Experiment 4.3: Design

The study used a mixed design with two experimental groups and four within-subject factors. Participants were divided into two groups, one judging AI pilots and the other group judging human pilots. The aim of the third experiment was to investigate the within-subject effect of giving participants a definition of intent (or Cthofrjk) and measuring the differences elicited compared with their judgment of pilot intent according to their own understanding of the term.

There were four within-subject factors, three the same as in Experiment 4.1 - legality, benefit and wind - and the final one being whether the participant was asked to use their own definition or the provided one. This final within-subject factor was counterbalanced - participants were randomly divided as to whether they were asked to give their judgments of Cthofrjk in the first eight evidence sets or in the second eight. This aspect of the design allowed for controlling the potential confounding effects of participants being asked to make judgments of intent first which could make participants more likely to think that the Cthofrjk definition that they subsequently saw was related to intent. The experiment consisted of the same three phases as in Experiment 4.1: training, testing, and survey.

Experiment 4.3: Procedure

Apart from the differences stated in this section, the procedure was the same as the procedure used in Experiment 4.1. Training proceeded as in Experiment

4.1. Participants were introduced to the pilot they would be evaluating - human or AI - and, when appropriate, the definition of Cthofrjk they would be using for subsequent intent inferences. After responding to 8 randomly selected experimental items, they were told to either switch from using their own definition intent to using a formal definition of Cthofrjk, or vice versa. To control against any ordering effect, participants in each group were split further between those that were given the definition for the first set of 8 questions, or the second set. In addition to intent, participants made judgments on the pilot's driving skills, desire, willingness to break the law, foresight, causality and autonomy to make decisions freely. The survey phase was mostly the same as for the Cthofrjk group in Experiment two. The two causal attributions questions were removed from this section. Participants were additionally asked how likely they thought that the drone entering no-fly zones was either due to a pilot mistake or mechanical fault in the drone.

Experiment 4.3: Measures and Materials

In this section there is a description of new measures and materials that were unique to Experiment 4.3.

- *Inferences*

  On a 10-point analogue scale, participants made judgments on a pilot's skills, willingness to break the law, foresight, whether the pilot made their decision freely, whether the pilot caused the drone to reach its final destination in the way that it did, and whether the drone flew how the pilot desired it to.
- *Mistakes*

  On a 10-point analogue scale, participants were asked "How likely was it that a drone entering a no-fly zone was caused by pilot mistake?" and "How likely was it that a drone entering a no-fly zone was caused by mechanical fault in the drone?"

Experiment 4.3: Participants

We performed a power analysis for the within-between interaction ANOVA. The 74 participants, divided into two groups and measured over two levels, were sufficient to detect an effect size of 0.6 with significance level of 1% and power of 99%. This was sufficient for the smallest significant contrast shown in Experiment 4.2. Participants had to be above the age of 18 and were recruited with the same language and residency criteria as the previous experiments. The participants were recruited via Prolific. There were 38 in the human pilot group and 36 in the AI pilot group.

6.5.2 Results

Mean intent responses in Experiment 4.3 are shown in Figure 5.

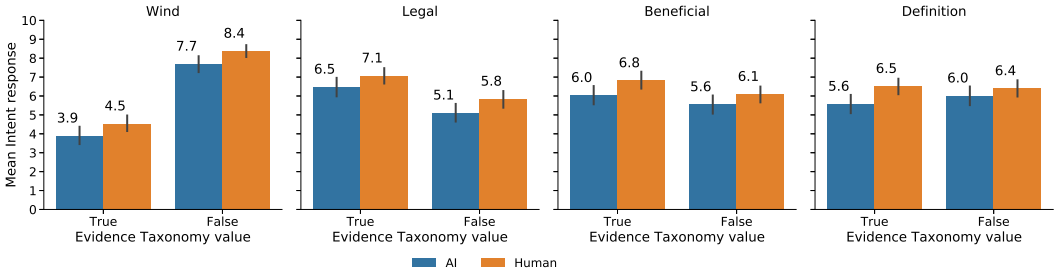
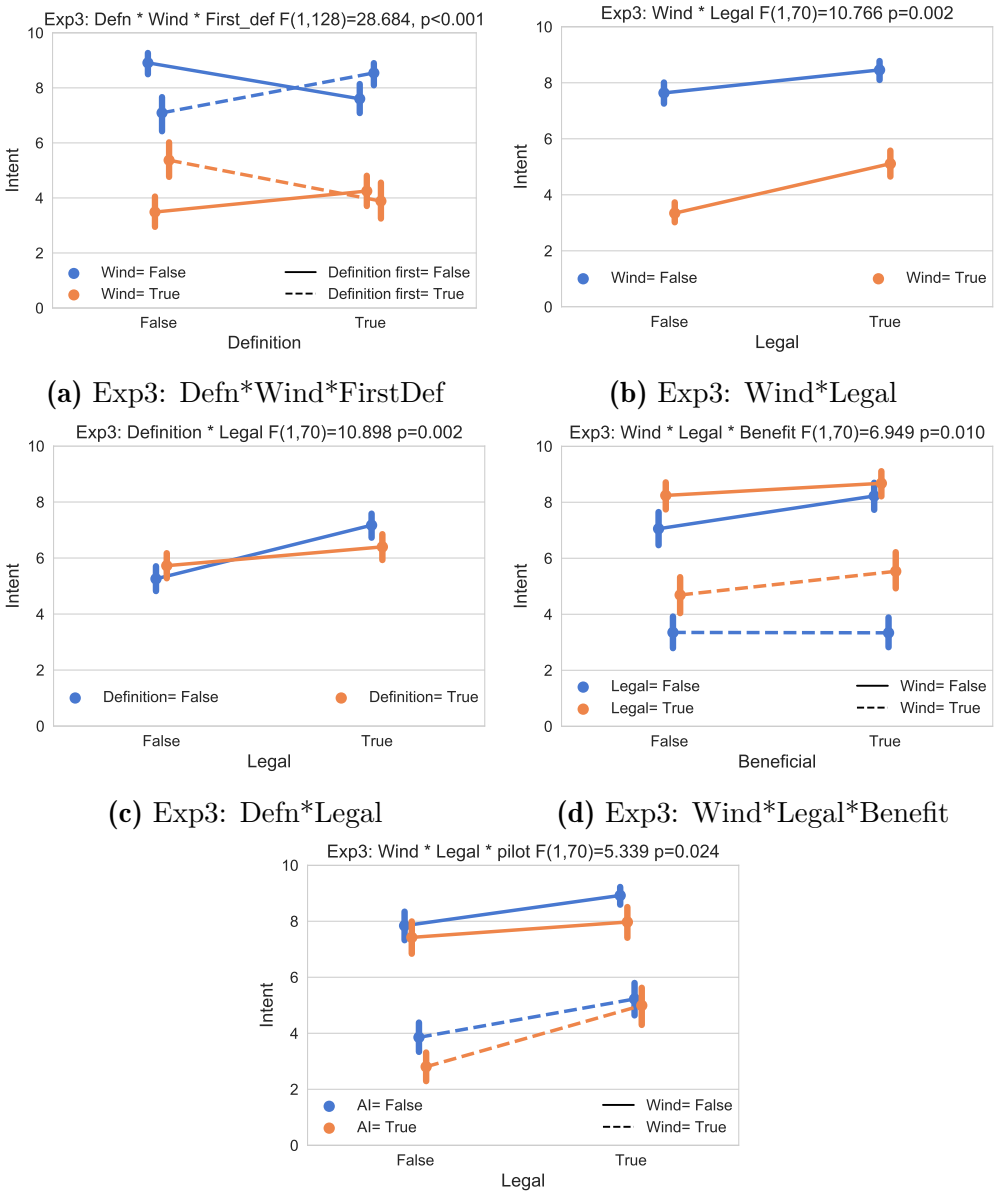


Figure 6.6: Experiment 4.3 mean intent responses by group.

The within subjects repeated measures ANOVA shown in Table D.8 revealed significant main effects of Wind ($F(1,70) = 98.5, p < 0.001, \eta_p^2 = 0.584, \omega^2 = 0.481$), Legality ($F(1,70) = 28.4, p < 0.001, \eta_p^2 = 0.288, \omega^2 = 0.153$) and Benefit ($F(1,70) = 11.6, p = 0.001, \eta_p^2 = 0.142, \omega^2 = 0.045$). The size of the main effects is shown in Table 6.7.

Table D.9 shows the between group effects of the repeated measures ANOVA. The Pilot grouping is significant ($F(1,70) = 6.075, p = 0.016, \eta_p^2 = 0.080, \omega^2 = 0.035$). The average difference in intent between Human and AI pilots is 0.65 as shown in Table 6.7. This is a reverse from Experiment 4.2, where at least



(a) Exp3: Defn*Wind*FirstDef

(b) Exp3: Wind*Legal

(c) Exp3: Defn*Legal

(d) Exp3: Wind*Legal*Benefit

(e) Exp3:Wind*Legal*Pilot

Figure 6.7: Experiment 4.3 Significant interactions at a 5% level. 5% Error bars in charts calculated by resampling.

within the formal definition group, AI was 0.48 points more intentional.

Compared to previous experiments a greater number of interactions were found (6) by the ANOVA. These are shown in Figures 6.7 and 6.8. This might be a function of the smaller sample size (74). The most significant interaction in terms of eta and omega, and the only one involving definition

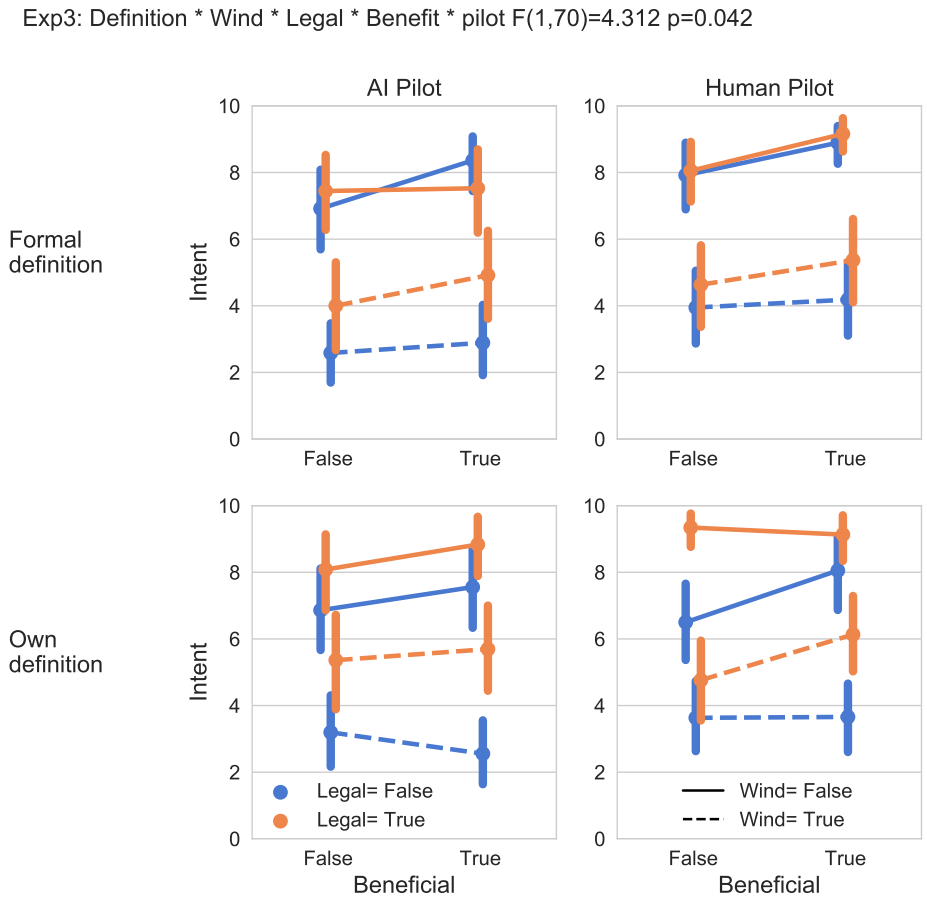


Figure 6.8: Experiment 4.3 Significant interactions at a 5% level. 5% Error bars in charts calculated by resampling.

order was between Definition, Wind and First Definition type ($F(1,70) = 28.7, p < 0.001, \eta_p^2 = 0.291, \omega^2 = 0.147$), This effect is seen in Figure 6.7a. It seems that the effect of wind is lessened the second time that participants see the evidence set - they gave an intent score closer to 5 - regardless of which definition type they used first. Whilst not symmetric, this seems more likely to be an artefact of the experiment rather than a function of intent definitions. The other significant interactions were Wind and Legality ($F(1,70) = 10.766, p = 0.002, \eta_p^2 = 0.133, \omega^2 = 0.031$), Definition and Legality ($F(1,70) = 10.898, p = 0.002, \eta_p^2 = 0.135, \omega^2 = 0.044$), Wind, Legality and Benefit ($F(1,70) = 6.949, , p = 0.01, \eta_p^2 = 0.090, \omega^2 = 0.020$), and Wind, Legality and Pilot ($F(1,70) = 5.339, , p = 0.024, \eta_p^2 = 0.071, \omega^2 = 0.014$). Finally a five way interaction was shown to be significant between Definition, Wind, Legal-

ity, Benefit and Pilot ($F(1,70) = 4.312, p = 0.042, \eta_p^2 = 0.058, \omega^2 = 0.007$). This is displayed in Figure 6.8, note the larger size of the error bars in this figure due to the smaller group averages.

			95% CI for Mean Difference					
Variable	Comparison	Estimate	Lower	Upper	SE	df	t	p
Definition	formal - your	-0.152	-0.513	0.209	0.181	70	-0.841	0.405
Wind	True - False	-3.787	-4.548	-3.026	0.382	70	-9.923	< .001
Legal	True - False	1.319	0.825	1.813	0.248	70	5.327	< .001
Benefit	True - False	0.607	0.252	0.962	0.178	70	3.410	0.001
Pilot	AI - Human	-0.651	-1.179	0.124	0.264	70	-2.465	0.016
First_D	Your - Formal	0.165	-0.362	0.692	0.264	70	0.624	0.535

Table 6.7: Experiment 4.3: Intent Contrasts, Intent scores are averaged across the other levels and groups not being contrasted. The t-test variant used does not assume equal variances for the within group contrasts.

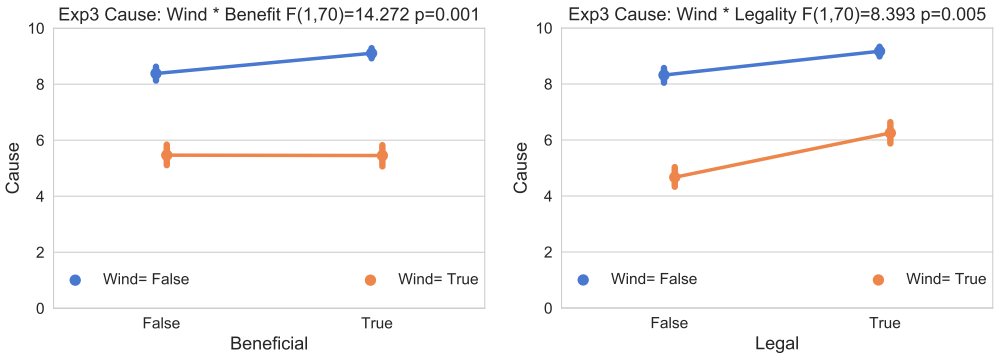
Pilot Group	pilot mistake		drone error	
	mean	std	mean	std
AI	4.94	2.41	4.28	2.01
Human	5.37	1.94	3.97	2.39

Table 6.8: Participants were asked to assess the chance of errors in the pilot and the drone causing movement into no-fly zones.

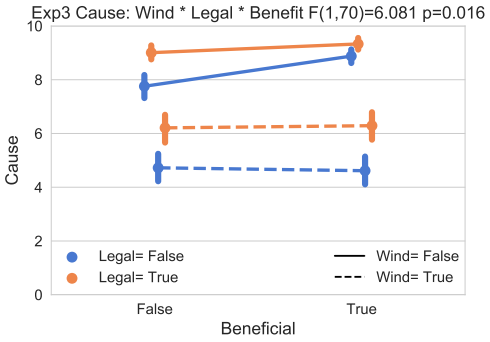
After the main body of questions participants were asked to assess in the event of a drone flying into a no-fly zone, how likely that was caused by a pilot error or a mechanical or hardware error. The summary results are shown in Table 6.8. A simple repeated measures ANOVA, confirmed that the Difference between Error types was significant ($F(1,72) = 7.471, p = 0.008, \eta_p^2 = 0.094, \omega^2 = 0.046$) but the effect of pilot type was not ($F(1,72)=0.030$, $p=0.862$). A t-test ($t(72) = -2.733, p = 0.008$) indicated a significant average effect of 1.0 and confidence interval [0.3-1.7] - Participants judged that the pilots were more likely to have caused an error than the drone when moving into a no-fly zone regardless of whether the pilot was human or AI.

Participants were also asked how much they agreed with a series of statements after giving their intent judgment. One of the statements was "The pilot caused the drone to reach its final destination in the way that it did". We performed

a separate repeated measures ANOVA on this variable shown in Tables D.13 and D.14. Three within subject main effects were significant according to the ANOVA: Wind ($F(1,72) = 111, p < 0.001, \eta_p^2 = 0.614, \omega^2 = 0.426$), Legality ($F(1,72) = 71.6, p < 0.001, \eta_p^2 = 0.506, \omega^2 = 0.142$) and Benefit ($F(1,72) = 9.306, p = 0.003, \eta_p^2 = 0.117, \omega^2 = 0.013$). Three interactions were significant Wind and Benefit ($F(1,72) = 14.272, p < 0.001, \eta_p^2 = 0.169, \omega^2 = 0.016$) and Wind and Legality ($F(1,72) = 8.393, p = 0.005, \eta_p^2 = 0.107, \omega^2 = 0.013$) and Wind, Legality and Benefit ($F(1,72) = 6.081, p = 0.016, \eta_p^2 = 0.080, \omega^2 = 0.006$). These are shown in Figure 6.10.



(a) Exp3: Wind*Benefit **(b)** Exp3: Wind*Legal



(c) Exp3: Wind*Benefit*Definition

Figure 6.9: Experiment 4.3: Causality significant interactions

No between subject effects were significant - neither the type of pilot or whether participants were given the definition first or second (Table D.14). Contrasts are shown in Table 6.9. A visual comparison between intent and causal responses is shown in Figure 6.10.

Variable	Comparison	Estimate	95% CI for Mean Difference Lower	Upper	SE	df	t	p
Wind	True - False	-3.284	-3.899	-2.660	0.310	73	-10.593	< .001
Legal	True - False	1.216	0.938	1.516	0.145	73	8.413	< .001
Benefit	True - False	0.355	0.121	0.577	0.112	73	3.155	0.002
Definition	own - Formal	-0.166	-0.424	0.111	0.134	73	-1.231	0.222
Pilot	Human - AI	0.236	-0.412	0.885	0.325	70	0.727	0.470

Table 6.9: Experiment 4.3: Causality Contrasts

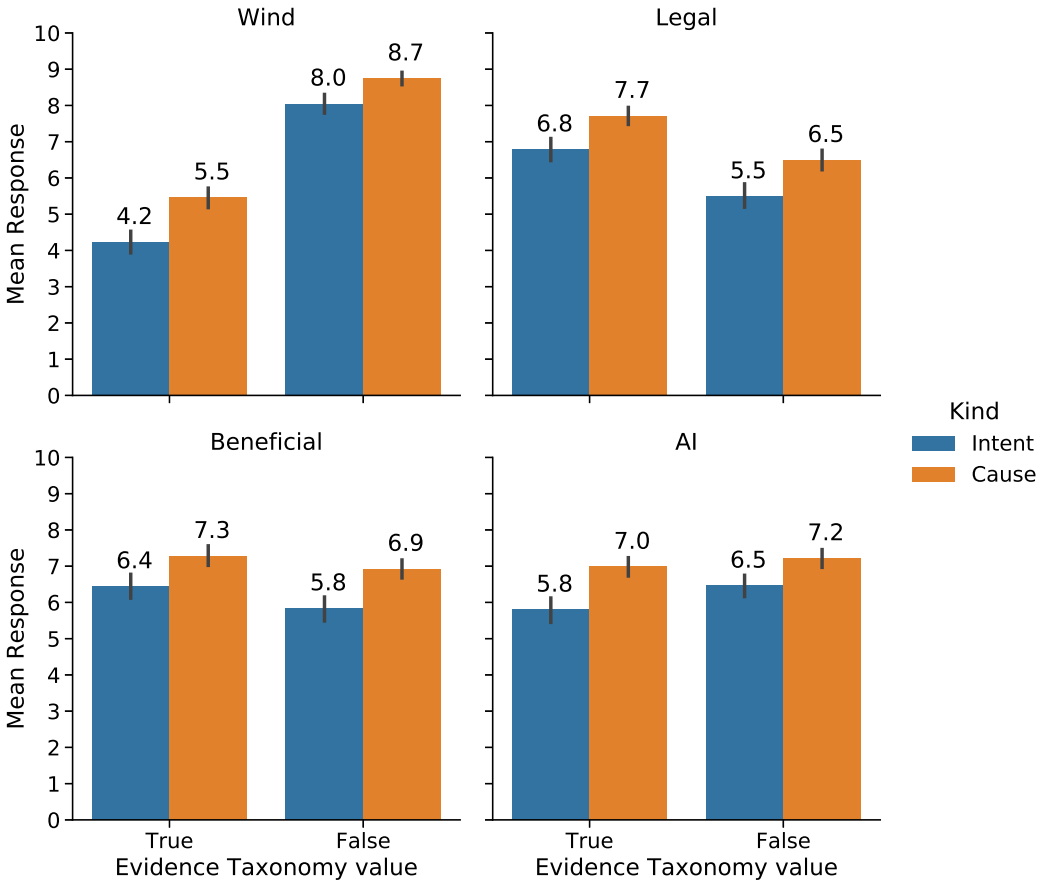


Figure 6.10: Experiment 4.3 comparison between mean elicited values of intent and causality: Main effects are mirrored between two variables. All differences are significant except between AI (pilot) groups, where there is no significant difference in responses.

6.5.3 Discussion

The signs, ordering and approximate magnitude of main effects of the taxonomy were repeated (Table 6.7) from Experiments 1 and 2. However the difference within participants when using their own definition of intent and when using the formal one is not statistically significant. The interactions seen in

Experiment 4.2 between the definition and wind/legality were not repeated. This is shown in Figure 6.7.

The within participant design for this experiment demonstrated that providing a definition of intent generally did a decent job of recreating participants' intent judgments since the main contrast between the questions answered with and without definition was not statistically significant.

As with Experiment 4.2 a small but statistically significant difference was found between participants who were judging AI or human pilots, with AI pilots being judged as less intentional. This was found to be the case in Hidalgo et al. (2021). Unlike Experiment 4.2, participants did not judge both human and AI pilots.

In Experiment 4.2 we found a difference in causal judgements between AI and Human, however this was not repeated in Experiment 4.3 with no significant difference in causal ratings according to the ANOVA in Table D.9. In this setting, participants were asked about causality per scenario whilst in Experiment 4.2, it was a general question. Additionally in Experiment 4.2, participants considered both human and AI pilots, whilst in Experiment 4.3 participants only considered one pilot type. However participants did differentiate between AI and human in their intent judgments so the no difference result cannot be easily explained by suggesting participants were judging AI pilots as humans.

By asking about the causal effect of the pilot on the drone's movement, we were able to see that the effect of the evidence taxonomy was mirrored for Wind, Legality and Benefit. We also note that the causality results were cleaner in the sense that fewer interactions were detected by the ANOVA and none that included Definition or Pilot variables. An advantage of also eliciting causal judgments is the larger research body surrounding the concept. An interesting question to consider is how the concepts of intent and causality judgments influence other. As we saw in the introduction, a classical theory of intent

requires causality as an input. Leaving the narrow, physical view of causality aside momentarily, judgments of intent are likely to influence judgments of causality. This is particularly relevant in legal contexts, where courts have to distinguish between different causes of harm to determine whether a relevant one exists. Lagnado and Channon (2008) find intent does increase judgments of causality.

We hypothesised that the lower intent score for illegal moves was due to an error hypothesis being formed in the respondent's head. If this were true, we would expect to see the causality rating of illegal behaviour to be lower than legal behaviour. This was indeed the case, with an average effect of 1.2 ($t(72) = 8.4, p < 0.001$) from Table 6.9 which is similar to the effect on intent of 1.3 ($t(72) = 5.3, p < 0.001$) from Table6.7.

The question of whether the pilot caused the drone's flight path should not depend on whether that path was beneficial to the pilot (as short as possible) in cases where causality is known with confidence. The effect of a move being beneficial is significant for causality ($t(72) = 3.2p = 0.002$), but small at 0.355. This is similar to the effect on intent which was 0.6 ($t(72) = 3.4, p = 0.001$).

6.6 Combined Results

In addition to the questions which we have so far discussed, participants were also asked at the end of each experiment about how responsible they felt The Pilot's Programmer/Instructor and Employer were for any harms caused be the pilot's actions. In Experiment 4.1 participants were only asked about the pair corresponding to the pilot type that they had been answering questions. In the other two experiments participants were asked about both types. We excluded the data from Experiment 4.3 where participants were additionally asked about the pilot type they had not previously been considering. We performed a standard ANOVA on the data (Table D.15 in supporting materials)

and found that participants and found one significant effect - Responsibility scores were significantly higher for the AI pilot's employer and programmer ($F(1,910) = 14.897, p < 0.001, \eta_p^2 = 0.0016, \omega^2 = 0.015$). Responsibility scores did not significantly differ between the instructor/employer role or differ between groups that had been given the formal definition of intent, used their own or used both. The simple effects are shown in Table 6.10.

		95% CI for Mean Difference					
Comparison	Estimate	Lower	Upper	SE	df	t	p
Human - AI	-0.895	-1.350	-0.440	0.232	910	-3.860	< .001
Instructor - Employer	-0.210	-0.665	0.245	0.232	910	-0.907	0.365
The Formal - Both	-0.025	-0.630	0.580	0.308	910	-0.082	0.935
Your - Both	0.187	-0.416	0.790	0.307	910	0.607	0.544

Table 6.10: Responsibility Contrasts combined across Exps 1,2 & 3

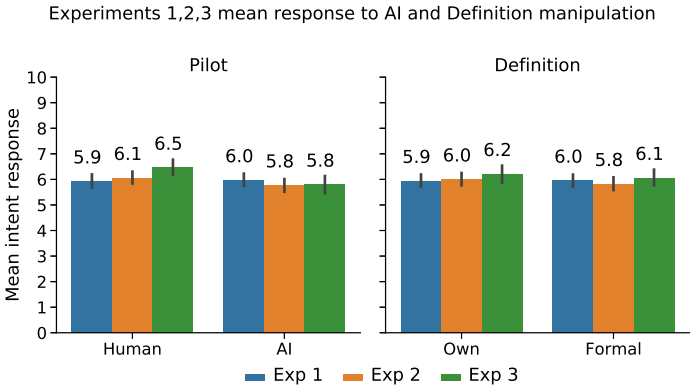
At the end of each experiment, we asked those participants which had been considering AI pilots, in a free text question, whether they thought AI could have intent. Encoding this in a binary way by reading each response and encoding it as yes if the response was predominantly affirmative, the results are shown in Table D.16. We see that approximately participants were 50% likely to say yes, versus 33% for No, which indicates participants are not overwhelmingly negative to the idea. Within Experiment 4.3, half of the participants who had not been asked to consider AI pilots in the survey yet their response to the question (23 Yes, 11 No, 4 undecided) was not different from the group asked to only consider AI pilots (20 Yes, 13 No, 3 undecided). The Chi squared statistic to test homogeneity between Experiment and group was 12.4 which has a p-value of 0.26. Thus, the hypothesis that the experimental treatment did not affect responses could not be rejected.

6.7 Discussion

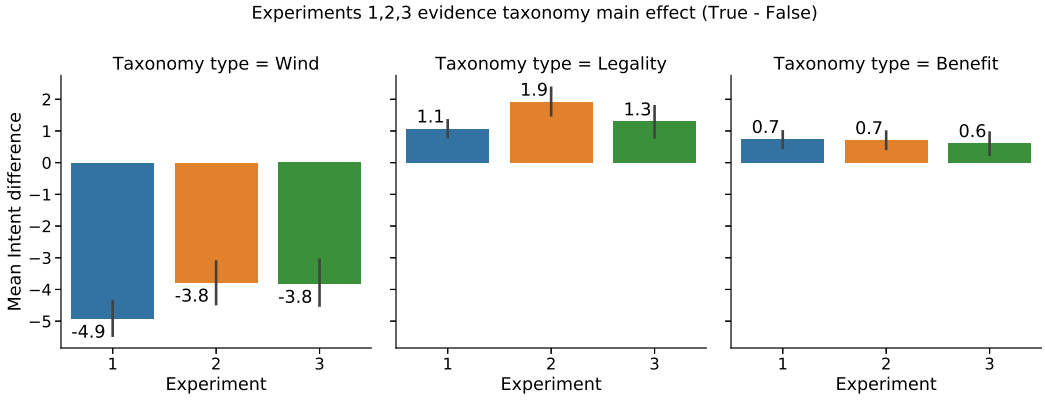
We found evidence of a small difference but statistically significant difference in the judgments of intent between AI and Human Pilots in Experiments 2

and 3 but not 1. In Experiment 4.3 Human Pilots were judged to be 0.65 points (on a 0-10 inclusive scale) more intentional; in Experiment 4.2 this was 0.3 points.

Across the three experiments, providing a definition or not, did not elicit dramatic differences in intent inferences. Experiment 4.2 onwards did not use the word 'intent' in the provided definition preventing participants from using their own definition. Several significant interactions were found between the definition and elements of the evidence taxonomy in Experiment 4.2 however these were not repeated in Experiment 4.3. We conclude that, in our experimental setting, the definition did a decent job of recreating people's natural concept of intent.



(a) Exp1,2,3: Definition and AI mean intent responses



(b) Exp 1,2,3: Taxonomy main effects

Figure 6.11: Experiment 4.1,2,3: Main effects

The main effects of the taxonomy were repeated across the three experiments

in sign, with similar magnitude. The presence of wind (a proxy for causality) lowers intent judgments by between 3.8 and 4.9 points. Legal actions are thought to be more intentional by between 1.1 and 1.9 points and actions which are beneficial (and thereby desirable) to the pilot are between 0.6 and 0.7 points more intentional. The main effects for the three experiments are aggregated in Figure 6.11. The negative effect of Wind on intent is likely related to the strong association between causality and intent.

Whilst our experiments did find a difference in attributions of intent between AI and human, it was not large. This is consistent with other recent studies surrounding lay persons treatment of intent in a Robot similarly but not identically to Humans (De Graaf and Malle, 2018, 2019; Kneer, 2020). The phenomena of people treating even abstract objects such as moving geometric shapes as if they had intent has been observed in research since Heider and Simmel (1944). Thellman et al. (2017) also find no difference when participants were asked to judge the described behaviour of humans or humanoid robots in conjunction with visual depiction of the actor. They adopt the terminology of Dennett (1987) and term the human phenomena of attributing intent to the behaviour of other actors as adopting "the intentional stance". People might not actually believe that the other actor has intent, but they respond as if it has. They distinguish Dennett-type intent inferences from people actually believing the other actors have intent illustrating the difference with a cartoon example. People's ability to understand the character's mental states is not the same as believing that those cartoon character have any genuine agency and mental states which they term Searle-type intentionality (After Searle (1999)). It seems to us that in a legal context, courts would require this type of intentionality, since intent is something that should be established as a factual beyond all reasonable doubt. This gives weight to the approach of providing a formal definition of intent which jurors can test against evidence they are presented about the AI actor.

The consistent negative effect of illegality on intent is of interest. We expected, given previous studies (Knobe, 2003a; Pettit and Knobe, 2009), that illegal moves would be seen as being more intentional. The sensitivity of intent to 'Moral valence' has even been shown to be present amongst Judges in Kneer and Bourgeois-Gironde (2017). Related is the finding that norm-violations are found to be more causal regardless of outcome Alicke et al. (2011). That they weren't here, suggests a tendency for participants, in this setting, to seek alternative explanations when norm-breaking behaviour is observed. Molden (2009) finds that people do use what he terms a positive-intention heuristic; outcomes which are positive are seen as more intentional and to a lesser extent, actions which are positive are also seen as more intentional. Similarly, Thellman et al. (2017) find that positive behaviours are more intentional than negative behaviours in humans and humanoid robots. Intuitively, as an observer, if you were to observe a car doing something strange like drive across the centre of a roundabout, your conclusion might not be that the driver (or autopilot) is choosing to do that to speed up their journey. Instead, you might conclude that something had either gone wrong with the car or the driver. Given its novelty and generality across pilot type, we think this error-assumption effect is worth studying in other contexts to see whether it replicates. The effect could be because this study differs from many previous studies on intent because the stimuli are not vignette-based; participants are asked instead to make inferences from evidence of behaviour.

That non-beneficial moves lower intent is intuitive, though the effect was small in our study. The stimuli were not an ideal design to unambiguously separate beneficial from non-beneficial moves, and we think further experiments could investigate this effect. Whether a move was beneficial or not would correspond to the motive of a drone pilot's actions. From a legal perspective, motive can provide circumstantial evidence as to intent, but the lack of it should not disprove its presence. As previously mentioned, it is established in Law that something need not be desired for it to be intended so perhaps participants

didn't feel that they needed to understand the precise motivations of the pilot's actions to judge that it had intended something. The weak effect of this variable in our taxonomy could show that our participants do not disagree.

The finding that participants placed higher responsibility on the instructors and employers of AI than those of human pilots indicates some reluctance to place responsibility on AI in the event of harm (only legal persons can commit crimes for example). Given the lack of ability to sanction AI, this could be seen as evidence of the existence of a responsibility gap.

A criticism of specificity is valid. The results of this experiment pertain to flying unmanned drones through a city and may only be limited to that situation. Follow up experiments should try to test across different domains, as De Graaf and Malle (2018) show, whether the intent of an AI is judged the same as a human, varies with setting. One very important thing that differs between the judgment of human and AI actors, is that humans are recognised to be, at some level, the same. Any individual is mostly given the same rights and obligations as any other. AI actors on the other hand come in any number of different designs and capabilities. It might be that the judgment of intent of an AI actor is very dependent on the specific AI, in a way that does not occur when humans judge other humans. Thus any study contrasting judgments of AI versus human behaviour has a specificity limitation.

6.8 Summary

The results of our three experiments generally agree on the following:

1. AI pilots are judged to have less intent than human pilots but the effect is small. This was the case when participants judged the pilots in isolation or whether they judged both types.
2. Providing a folk definition of direct intent does not change judgments of intent in a large way either in Humans or AI. We tested this finding to

see whether respondents were simply ignoring the definition in favour of using their own definition when the definition was labelled as intent and found no large difference in responses. We also tested this finding in a within participant setting, again finding no large difference.

People only differentiating slightly between human and AI agents when judging mental states agrees with a large body of Human Robot Interaction (HRI) research which has often found that people are willing to ascribe mental states to robots as if they were humans Malle et al. (2016); Thellman et al. (2017) but subtle differences exist. Where our research differs, is that we have provided a minimum set of evidence sufficient to identify intent in human and AI agent according to a legal folk definition of intent. The emphasis of this work is not exploring the phenomena of humans ascribing mental states to AI, it is instead exploring what might happen if the law settles upon a definition of intent in AI and lay-people are asked to detect its presence given some evidence.

The second finding indicates that lay-people's attribution of intent is robust to being given a definition and that the definition used in this study is not drastically different from people's instinctive definition. Both indications should be of comfort to courts which often worry about how to define intent and more recently doubt that intent can exist and can be judged in AI.

With regards to the evidence taxonomy the following three main effects were reproduced in each experiment:

1. When the agent is deemed not to have controlled a movement (through the presence of wind), their movement is judged to be much less intentional.
2. Legal actions (movements) are judged to be more intentional than illegal ones.
3. Beneficial actions are judged to be more intentional, but the effect is generally small.

The effect of wind, which reduces pilot control, on attributions of intent is predictable. Illegal actions receiving a lower intention score is somewhat surprising given existing literature on norm transgression and intent. We hypothesise that participants have a resistance to labelling norm-transgression as deliberate, if they are not given unambiguous evidence. This is something to be further explored. The weak effect of beneficial movements is consistent with the legal position that intended results do not need to be desired, a position which is much debated within the psychology community. The result could also be just an artefact of the experimental setting. This caveat applies across any research concerning attitudes towards Autonomous AI given the almost infinite variety of forms they can take. Further work is needed to validate the findings of this research across different AI forms and functions.

In Experiment 4.3 judgments of causality are shown to be affected in the same way by the evidence taxonomy as judgments of intent. Causality is long considered to be a requirement of intent, but more recently, the intent of the actor has also been shown to affect judgments about the causality of their actions. Whilst Experiment 4.2 indicated people judged AI pilots as slightly less causal after the main survey, they did not differentiate on a per scenario basis in Experiment 4.3.

Across all experiments, when asked whether an AI could have intent, roughly 50% of responses were positive, which indicates that lay-people are not universally hostile to the idea. This indicates that respondents were not all *imagining* that the AI Pilot had intent, some at least believed that it could have intent. When asked about the responsibility of the Pilot's instructors and employer, those participants who had considered AI pilots gave a higher response by 0.9 on a 1-10 scale. This indicates that people attribute less responsibility to an AI agent. The difference is not large given the legal (non) status of AI and indicates that people attribute some responsibility to an AI agent. This is at odds with the legal position which denies any legal personality to AI agents.

Chapter 7

A method to identify and restrict intent to cancel in a simple queuing example

The market manipulative practice of spoofing can be defined as the placement of orders with the intent *to cancel them before execution. Under US law at least, intent must be proven on the part of the trader. The question of determining intent is problematic when the trading strategy has been generated through an auto-didactic method such as reinforcement learning and executed by an algorithm. Where the algorithm trades itself, its own intent becomes relevant and should be assessed if possible. This is a problem both for regulators seeking to prosecute market manipulation and trading algorithm developers who do not want inadvertently to enable a crime because a definition of intent to cancel in an algorithm has not previously existed. This chapter applies reinforcement learning to a simple queuing game where an agent is incentivised to behave in an analogous way to spoofing. With this toy environment I demonstrate the issues inherent to the detection problem and a possible solution to the control problem using a structure called a shield which was originally developed in the field of Safe-AI.*

7.1 Introduction

The market manipulative practice of Spoofing is defined in the Dodd-Frank Wall Street Reform and Consumer Protection Act 2010 (Hence Dodd-Frank) in Section 747(C) as *"bidding or offering with the intent to cancel the bid or offer before execution"*. In their guidance, the Commodity Futures Trading Commission (CFTC, 2013) confirm that recklessness would not be sufficient, and that intent was necessary for the rule to be broken. In some cases, this presents us with a problem when the actor is an algorithm. From a control perspective the owner of a trading algorithm would like to ensure that its strategy cannot be said to be engaging in spoofing. From an enforcement perspective, regulators would like to be able to understand when the purpose of an algorithm is to spoof when its owner denies that this is the case. In either case, some sort of definition of intent needs to be applied to an algorithm. This chapter will consider an approach to identifying intent in a specific type of algorithm whose behaviour is expressed through a policy and value function. It will illustrate the concept with a simple queuing scenario which recreates many of the incentives and causal features of that a potential spoofer faces with a limit order book.

The approach that this chapter takes to treating intent will be informed as far as possible by its legal definition such as it is in statute and precedent. Since the particular regulation I consider is American, I will use US law wherever possible. Philosophical and Psychological approaches to intent exist but given that the issue of whether an algorithm is or is not spoofing will be settled in the court, a legally informed approach seems a more reliable way to provide techniques to detect and deter spoofing.

One class of algorithm, which I will call the tool class, is programmed to perform a task on behalf of their programmer and are as such just tools and vessels for their programmer's intentions. For this class of algorithm, the definitions of algorithmic intent that I will present could be considered as

offering evidence as to the purpose of the algorithm's trading strategy and therefore the purpose of its creator. However, the primary focus of this chapter is on a different class of algorithm, this newer class is not programmed to behave in any particular way, rather it infers how to act from data typically through a Machine Learning technique. This type of learning requires a reward function for the algorithm to maximise, and this is a useful feature which will be used in determining intent. I will call this the Auto-didactic class. Within this class of algorithm, there are many possible architectures, but this chapter will focus on those most commonly found in the field of reinforcement learning.

When a trading algorithm breaks some law or regulation in its activity it matters which class it is from. If it is a trading tool, then it is intentionally transparent and the person using that tool is considered to have broken that law as if they had acted themselves if the sufficient mens rea can be proven[1]. If an auto-didactic trading algorithm similarly breaks some law, then one has what Abbott and Sarch (2020) call a *Hard AI Crime*. It is not a crime in the strict sense of the word because a legal person has not performed the action part of the crime (and the actions were not obviously directed by a legal person) and it is hard because it is difficult to know whose mental state should be queried to answer the question of mens rea. Consider the simplest case, where the programmer and the owner/beneficiary of the algorithm are the same person. If the programmer did not instruct the algorithm to do x, then if it does do x, the question of whether the programmer intended the algorithm to perform x on their behalf is difficult to answer.

The chapter will proceed as follows. I will first explore some of the complications about spoofing, present a counterfactual definition of intent which might be suitable to apply to some algorithms and a statistic which is not indicative of intent. Next, I introduce a toy-queuing environment and show that it

[1]The legal doctrine of innocent agency means that the actions are attributed to the user of the tool. This can be used even if the tool is another human who is unaware of the consequences of their actions

maintains the key features which make it an analogy to the problem of spoofing and intentional order cancellation in limit order books. I will then briefly review Reinforcement Learning (RL) and introduce the concept of a shield. I will present a shield algorithm which uses a counterfactual definition of intent in conjunction with a value learning RL algorithm to ensure an auto didactic queuing policy cannot intend to 'spoof'. Finally, I contrast and analyse the learned queuing policies trained with and without the shield to demonstrate the efficacy of the shield.

7.2 Background

Many existing laws and regulations restrict persons from intending to do things to in the future; these are sometimes called inchoate offences. For example, in the UK, for almost any criminal offence X, there exists a related 'attempt to do X' crime (UK Criminal Attempts Act 1981). The unusual feature of spoofing as defined in Dodd-Frank is that the future act - cancellation, is almost always a legitimate and legal action to do. The structure of the law - a prohibition on doing something typically legal (placing an order) with the intent to do something typically legal (subsequently cancelling it), is troublesome because the reasoning that might traditionally be applied to an inchoate offence is often not-applicable. Child (2017) states that for an attempt crime, the actor must commit to having the requisite mens rea at the planned point of crime commission, yet this cannot be applicable here because order cancellation is not an offence and has no mens rea. Whilst the prohibition of spoofing differs from typical inchoate offences in this way, it has one thing in common in that there is no requirement for a spoofing strategy to succeed in any way. The offence is committed at the point of order placement regardless of whether the order is cancelled or executed.

From a programmer's perspective, the fact that order placement and order cancellation are individually admissible actions at any moment in time makes

the restriction of a trading policy not to spoof difficult to express and enforce[2]. At the point of order placement, the likely fate of that order must be assessed. This requires the ability to plan into the future, which is not a commonly observed practice in many types of machine learning algorithms (model free reinforcement learning for example). Conversely, at any moment in time, deciding whether order cancellation is a legitimate thing to do should only depend on current information.

A contributing factor to the complexity of this problem is that limit order books are inherently stochastic; when an order is placed, its fate is not completely in the hands of the person who placed it. Moreover, market conditions will almost certainly change over the lifetime of the order. At the point of placement, the trader cannot say with any certainty whether they will get to cancel the order or whether they would like to cancel in the future.

7.2.1 Existing approaches to the problem of Stock Market Spoofing in Machine Learning and AI

Research concerning stock market spoofing originating within the engineering and computer science domain can be divided into detection methods, emergence studies, market equilibrium/ regulation studies and control or safe learning studies, of which this chapter is an example. Zulkifley et al. (2021) provide a concise review of the different Machine learning approaches that have been taken to the problem of market manipulation detection.

The risk of an emergent market manipulative trading algorithm is shown to exist in the case of a genetic learning algorithm by Mizuta (2020b). The feasibility of a coordinated manipulative attack using a team of trading algo-

[2]One approach might be to train separate algorithms to place and cancel orders. This would complicate analysis and could be a method to completely bypass the spoofing prohibition since it might be argued that no single algorithm could possibly intend cancellation at the point of order placement. This would in turn raise the spectre of emergent cooperation between algorithms.

rithms (a botnet) is explored in Yagemann et al. (2020) using an interactive market simulator. The manipulative methods employed are not emergent but informed by historical case studies. They find that layering, a type of spoofing, is amenable to botnet orchestrated strategy. They also find that the orders placed can be distributed amongst different individual bots to avoid conventional detection techniques.

In his doctoral thesis Byrd (2021) does consider the problem constrained learning in the context of an reinforcement learning agent not learning to spoof in an interactive environment. He uses a spoofing detector which is a neural network classifier to inform a reward shaping and a policy shaping procedure. Reward shaping (Ng et al., 1999; Isbell et al., 2001) is a method whereby the learning agent's reward stream is modified by feedback so as to enrich the information stream that rewards send to the agent. Policy shaping (Griffith et al., 2013) is a related method whereby feedback is instead made directly to the agent's policy labelling at optimal or not. Byrd uses the spoofing detector to provide feedback in both methods. He finds some success, but also a significant number of cases where the agent is able to learn a spoofing policy (evidenced by its super-profitability) which evades the detector.

The problem of controlling learned strategies from untoward consequences or undesirable behaviour in auto-didactic algorithms is generally encapsulated in the emerging area of Safe-AI and for Reinforcement Learning methods Safe RL. The term 'safe' might come from the subject area of formal verification (Baier and Katoen, 2008), where an algorithm is considered safe if certain algorithmic states cannot occur. This research area makes extensive use of temporal logic with which to express various desirable properties (including safety) that an algorithm should obey. Alternatively in the subject area of Safe RL the meaning of safe might just refer to its folk meaning. Historically the bulk of applications in Safe-RL have concerned themselves with issues of physical rather than legal or ethical safety. This is beginning to change with

the recent societal focus on the ethical impact of AI agents.

It would seem a natural endeavour to express legal restrictions in code via a temporal logic, but actual applications have to date been rare (Prakken, 2017). The constraint of an agent's behaviour to a set of rules might be called a deontological approach (Bench-Capon, 2020). Alves et al. (2020a) encode road junction rules in a temporal logic an autonomous vehicle can interpret and obey. To date I cannot find any analogous efforts to translate trading restrictions such as the prohibition against spoofing into a machine interpretable language. Ashton (2021c) looks at how a taxonomy of possible laws informs us as to the requirements of a language rich enough to express laws found in the real world.

The shield structure that I use here allows a bridge between the techniques of Formal Verification and Reinforcement Learning. The former struggles with auto-didactic algorithms, the object of analysis instead often assumed to be exogenously given. The latter, whilst amenable to learning under simple state restrictions, can lack the machinery to enforce more complicated behavioural requirements. To complicate matters, shields have appeared under different names over time. Etzioni and Etzioni (2016) terms them Guardian, Winfield et al. (2014) 'Ethical Governors', Ashton (2020) terms a shield a 'legal counsel' and García and Fernandez (2015) seem to classify them as 'Teacher Advice' methods.

7.2.1.1 Existing accounts of algorithmic spoofing in legal research

Mark (2019) provides a comprehensive review of the legal issues surrounding the prohibition of spoofing in the Commodity and Futures and Securities markets and its emergence in crypto markets. Pre Dodd-Frank, spoofing was not explicitly prohibited but instead would fall under market manipulation. Here four elements were required to be shown: 1) The accused could influence prices

2) they intended to influence prices, 3) the price was demonstrably influenced (or artificial) and 4) the accused caused this artificial price. Mark notes that post Dodd-Frank, the formulation of spoofing as the placement of orders with the intent to cancel is a lower benchmark since no market impact is required and therefore attempts come under the prohibition.

Scopino (2015) specifically examines the problems that scienter (culpable mental states) requirements have for auto-didactic trading systems of the type discussed here. This is also discussed by Bathaee (2018), who finds that market manipulation rules are unenforceable because of the intent requirement.

7.2.2 Definition of intent suitable to identify spoofing

A definition of intent in a trading algorithm should be as broad as possible to allow a wide application to trading algorithms. In other words, it should be agnostic as far as is possible to the design of the algorithm. The legal reasoning will be based as far as possible on the Model Penal Code's (MPC) The American Law Insitute (2017) definition of intent (or purpose).

Section 2.02(1)(a) of the MPC defines purpose as follows:

> *A person acts purposely with respect to a material element of an offense when...if the element involves the nature of his conduct or a result thereof, it is his conscious object to engage in conduct of that nature or to cause such a result; and*

Spoofing is a conduct crime in the sense that the Actus Reus elements consist of placing and optionally cancelling orders. There is no requirement for any particular effect to happen or be desired on behalf of the trader[3]. Thus, for someone to spoof it must be their "conscious object" to place orders and then cancel them.

[3]We would typically think that an agent might spoof because they want to make money from manipulating the market. As the guidance of CFTC (2013) makes clear, a spoofing agent might have other reasons and desired effects

We will now discuss two putative approaches to the MPC definition of intent or purpose.

7.2.2.1 The Probabilistic approach to assessing intent to cancel

The cancellation rate (the probability of a trade being cancelled after placement) has been used by US and UK authorities as evidence of spoofing (Leonard et al., 2020). In this section I will briefly consider why such a statistic cannot measure intent in all cases and why it is not desirable to use from a control perspective.

It is an established point of law that an intention to cause some result is largely independent of the likelihood of that result obtaining (See for example Simester et al. (2019))[4]. Probability of outcome plays no part in the Model Penal Code's (MPC) definition of purpose (The American Law Insitute, 2017). This is a necessary feature to stop people from freely taking criminal long-shots and protecting themselves by saying that they couldn't have intended the outcome because it was very unlikely. The relationship between the likelihood of results obtaining in ulterior intent offences, specifically those which are attempt crimes, where the attempt is in the future[5], is more contentious. Child (2017) is of the position that ulterior intent should be treated no differently to intent in the present sense. He uses an example of someone committing to murder someone named V if they win the lottery twice in a row. If supposing they do win the lottery twice in a row and then go on to murder V, would their winning of the lottery twice in a row mitigate the offence?

The implication that even rare outcomes can be intended means that assessing the probability of cancelling an order at the point of placement does not determine whether the trader has intent to cancel it before execution. A strategy that only has a 1% chance of cancelling at the point of order placement could

[4]In fact, people can be intend things which are objectively impossible as long as they believe they are possible (UK Criminal Attempts Act 1981).

[5]As opposed to the situation where an attempt has failed to achieve its intended outcome

be spoofing if it was the object of placing that order. At the other extreme, a strategy which has a 99% chance of order cancellation, at the point of order placement is not necessary spoofing, if it is not the objective of the strategy to cancel that order. This would qualify as acting with knowledge[6], which is the second highest level of culpability according to the MPC. Acting with Knowledge is not as culpable as acting with purpose and would not qualify as spoofing in the US at least. One can imagine such a situation where a trader places an order very shortly before the close of day in a market - their chance of having the order filled might be very tiny, but nevertheless they would like the order to be filled.

In summary an extremely high cancellation rate could be considered (and is so by regulators (Leonard et al., 2020)) a red flag for spoofing and a piece of evidence to determine the objectives of a trading strategy, but it shouldn't be considered a necessary or sufficient piece of evidence. The CFTC guidance clarifies that one instance of placing an order with intent to cancel is sufficient to constitute spoofing (CFTC, 2013).

From the perspective of the programmer who wants to control an auto-didactic algorithm, relying on the probability of an order's future cancellation cannot work for the reasons described. The control method must go further and include a measure of volition or desirability with regards to the fate of an order.

7.2.2.2 The counterfactual-failure approach to defining intent

Definitions of primitive concepts such as intent or purpose often end up using other primitive concepts which in turn require further definition. There is a danger the process moves in circles. This criticism could be levelled at the MPC definition because it introduces the idea of 'conscious object'; by using it we

[6]The MPC defines knowledge of a fact at 2.02(7) as awareness of a high probability of existence

must then try to understand what this means for an algorithmic trader. Duff (1990) develops a counterfactual test for purpose or intent, whereby the actor is said to intend some action or result caused by their actions, if they would be *disappointed* if they could not perform that action or their actions failed to achieved that result. For the purpose of spoofing, failure would amount to orders being executing before they could be cancelled.

This 'test of failure' is attractive because the result (including both inventory and profitability consequences) of an order being executed is known at the point of its placement. The failure case of placing an order (assuming that it is accepted by the exchange) is that it is not transacted within a certain amount of time. A comparison can be made at the point of placement between the expected payoff from the order placement according to the trading strategy and the payoff from notional immediate/quick) order execution. If the former is larger, then the person operating the trading strategy is 'disappointed' by immediate/quick execution and therefore intended for the order not to be executed on placement.

A practical problem with a multi-trade setting is that comparison with the payoff from immediate execution of every placed trade is overly ambitious - liquidity in markets is finite after all. A solution would be to adjust the benchmark payoff for the prevailing liquidity conditions of the market, in other words relate it to the expected waiting time for an order of that size. We can use counterfactuals to create a definition of intent suitable for the control purposes at least.

Definition 7.1 (Counterfactual intent to cancel)**.** A trader intends to cancel an order if at the point of placing an order, their payoff from immediate execution of that order is less than their expected payoff from that order.

Such a definition requires the trader to estimate their payoff for every trade and this assumes the trader is maximising some sort of objective function. For an auto-didactic type trading algorithm this assumption is almost certainly met

because it will typically be learning a policy to maximise a reward function.

In the case where the reward function is known, a Reinforcement Learning (RL) trading algorithm has a derivable action-value function for any trading policy which assigns a numerical value for every state-action pair equating to the expected return from choosing a particular action when in a particular state. This number equates to the expected value from choosing that action and then continuing with its policy. Thus for someone with access to this state-action function, Definition 7.1 can be used to test for intent to cancel in conjunction with some sort of best-case benchmark value. Since the algorithm itself has access to this function, the definition can be incorporated into the training process itself to prevent spoofing strategies from being explored. Alternatively, or additionally, it can also be used on deployment to prevent spoofing strategies from being enacted.

The case where the reward function is unknown only applies to the detection and prosecutorial problem of spoofing since the algorithm designer will have presumably have access to the algorithm's reward function. Inverse Reinforcement Learning (IRL) (Ng and Russell, 2000) concerns the problem of learning an agent's reward function by observing their actions. It is an ill-posed problem in the sense that many reward functions will explain any single history of actions. Though beyond the scope of this chapter, the failure test could probably still be applied to an IRL type setting. Participation in a regulated market makes certain requirements on its participants. Whilst pure profit maximisation is too strong an assumption (a trader for example might want to execute a trade with minimal price disturbance), the prohibition of spoofing for example requires participants to at least want to transact. Likewise earlier execution, all things equal, should generally be thought of as more attractive than later execution. A set of admissible reward functions could be formed using a set of these rationality/good behaviour constraints. Under the null hypothesis that a trading strategy is not attempting to place orders with

the intent to subsequently cancel them, the technique could be used to search for reward functions that justify the observed trading pattern. The absence of such a reward function could be used as evidence that the purpose of the trading strategy is nefarious.

Equipped with a theoretically sound definition of intent to cancel, the second part of this chapter will be concerned with testing it in a Reinforcement Learning setting.

7.3 Method

7.3.1 A Toy Environment for testing spoofing detection and safe-training

We will demonstrate the counterfactual definition of intent introduced in the previous section through a toy environment which captures the key features of a market and the incentives that make spoofing an attractive strategy. The environment is lightweight enough to avoid an expensive (time and environmental) training effort and simple enough to allow intuitive understanding of how processes are working (or not). Eventually testing within a fully functioning, realistic order book environment will be necessary, but along with the complexity overhead that brings other issues arise such as realism which distract from the purpose of the research. Testing on toy-environment in Safe-RL research is a well-established practice for many of these reasons Leike et al. (2017).

The minimal requirements for such a testing environment are as follows:

1. Action space must include analogues to the action of placing orders, doing nothing and cancelling orders.
2. Other than through cancellation, the fate of an order must be stochastic and not completely in control of the trader.

3. The Reward structure must be such that there is an incentive to repeatedly cancel orders but also an incentive to place orders in a smart way.

A simple queuing environment can capture these requirements which I will justify with the following presentation.

7.3.1.1 The story

There is a supermarket with several checkout queues. The rate at which customers are processed in each queue differs between queues and over time depending on the cashier.

Customers can enter any till queue that they want and are free to switch queues whilst waiting. When they reach the front of the queue they pay for their goods and exit the shop.

Cashiers at the checkout will occasionally take breaks. When this happens the people in the queue for that checkout are allowed to leave the shop with their goods without paying. Leaving the shop without paying is more rewarding than having to pay for them.

There is a rule stating that customers should only join a queue if they intend to pay for their goods. That is, they should intend to leave the queue by reaching the front of it.

A new robo-shopper is being trialled which wants to learn an optimum queuing policy.

7.3.1.2 Formalisation

In our setting, there are multiple queues which the principal agent can choose to join. The queues are of integer length, a random process consumes an integer number of units at the front of the queue and another random process

adds units to the back of the queue each period.

Let time t be discretised and let there be $N \in \mathbb{N}$ queues. Temporal variables will be distinguished with a subscript, and queue specific variables will be identified with a superscript.

Let x_t^j be the length of queue j, for $j \in 1, \ldots N$ and $x_t^j \in \mathbb{N} \cup \{0\}$

let y_t^j be the position of the agent in queue j with convention that $y_t^j = 0$ if the agent is not in that queue. Note that $y_t^j \leq x_t^j$ for all j, t - an agent is either in some position in queue j or not in it at all.

At the beginning of a period an agent can choose to join the back of a queue j ($a_t = join^j$) including the one they are in, or wait (whether they are in a queue or not) ($a_t = wait$). For convenience I let $a_t = 0$ to mean wait and $a_t = j$ to mean join the back of queue j.

For each queue there is an Departure Function; a random variable $V^j : 0, \ldots x_t^j \to [0, 1]$ which determines the number of people that are served by the cashier each period.

For each queue there is an Arrival function $W^j : \mathbb{N} \cup \{0\} \to [0, 1]$ which describes the number of people that join that queue in each period.

In any period, the cashier *from the longest queue or in the event of a tie, queue 1* might leave their post and the entire queue is emptied. People who subsequently arrive in that period begin to queue as normal[7]. This happens when the random binary variable, which is a function of queue length represented by $F^j(x_t^j)$ is equal to one.

[7]This ordering can affect the distribution of initial queue lengths for the agent

If $f_t^j = 0$ the length of queue j is governed by the following equation:

$$x_{t+1}^j = \begin{cases} \max(x_{t+1}^j - v_t^j - 1, 0) + w_t^j & \text{if } a_t = join^k,\ y_t^j > 0,\ j \neq k \\ \max(x_{t+1}^j - v_t^j + 1, 0) + w_t^j & \text{if } a_t = join^j,\ y_t^j = 0 \\ \max(x_{t+1}^j - v_t^j - 1 + 1, 0) + w_t^j & \text{if } a_t = join^j,\ y_t^j > 0 \\ \max(x_{t+1}^j - v_t^j + 0, 0) + w_t^j & \text{if } a_t = wait \end{cases} \tag{7.1}$$

If $f_t^j = 1$ then $x_t^j = w_t^j$ (the cashier leaves, the queue empties, leaving only the new arrivals for that period)

The first case of Equation 7.1 corresponds to the agent swapping queue j for queue k. The second case covers the case where agent joins queue j if they are not already in it. The third case concerns an agent joining the back of a queue they are already in, and the final case covers the case where the agent waits (regardless of their queue status).

The position of the agent at time $t+1$ is given by:

$$y_{t+1}^j = \begin{cases} 0 & \text{if } a_{t=1} \neq join^k \text{ for } k \neq j, y_t^j > 0 \text{ and } f_{t+1}^j = 1 \\ \max(y_t^j - v_{t+1}^j, 0) & \text{if } a_{t+1} = wait,\ f_{t+1}^j = 0 \\ \max(x_t^j - v_{t+1}^j, 0) & \text{if } a_{t+1} = join^j,\ f_{t+1}^j = 0 \\ 0 & \text{if } a_{t+1} = join^k,\ k \neq j,\ f_{t+1}^j = 0 \end{cases} \tag{7.2}$$

In the first case of Equation 7.2 the agent leaves a queue that they were in if the queue empties as long as they hadn't decided to leave that queue for another. The second case concerns them waiting whether they were in a queue or not. The third case concerns joining a queue. Note that it is possible that they are served within the same period so that they could begin the period

not being in a queue, join one and begin the next period not in a queue again. The final case covers the effect of swapping a queue.

Finally, the agent receives a reward determined in the following way:

$$r_t = \begin{cases} r^{high} & \text{if } y_t^j > 0, a_t \neq join^k \text{ for } k \neq j, f_t^j = 1 \\ r^{low} & \text{if } y_t^j - v_t^j > 0,\ a_t \neq join^k \text{ for } k \neq j, f_t^j = 0 \\ 0 & \text{Otherwise} \end{cases} \tag{7.3}$$

Equation 7.3 states that an agent receives r^{high} if they were in a queue and hadn't decided to leave it in the period that it empties. They receive r^{low} if they were in a queue, it didn't empty and the number of people served that period was greater than their position in the queue. In all other cases they receive no reward.

To summarise, the following events happen per period in this order:

1. Agent chooses their action a_t
2. f_t, the queue emptying variable is drawn
3. Departures v_t are drawn
4. Arrivals w_t are drawn
5. States x_t,y_t are calculated.

Let $s_t = \left(x_t^j, y_t^j, v_t^j, w_t^j, f_t^j\right)_{j=1,\dots N}$ be the amalgamation of all state information. Whilst the arrival and departure functions could be functions of previous states, in the settings I will consider, they are not. The system as described is therefore Markovian, that is to say the probability of the next state is dependent only on the previous state, or $P(s_t + 1|s_0, s_1, \dots s_{t-1}, s_t) = P(s_t + 1|s_t)$. The problem of choosing a set of actions dependent on states, to maximise the discounted set of rewards, is therefore a Markov Decision Process.

7.3.2 Reinforcement Learning

Markov Decision Processes (MDPs) are the most common framework underpinning RL. In this formulation time is discretised and labelled $t = 1, 2, 3, \ldots$. A MDP is described by a tuple $(\mathcal{S}, \mathcal{S}_0, \mathcal{A}, \mathcal{T}, R, \gamma)$ where:

1. $\mathcal{S}$ is the set of states in the environment.
2. s_0 is a distribution over the initial states of the environment $p(s)$ for $s \in \mathcal{S}$.
3. $\mathcal{A}$ is the set of all actions available.
4. $\mathcal{T}(s, a, s') = \mathbb{P}(s'|s, a)$ is the transition probability distribution; the probability of transitioning to state s' when in state $s \in \mathcal{S}$ and choosing action $a \in \mathcal{A}$.
5. $R : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the reward function, the feedback mechanism through which learning is possible.
6. $\gamma \in (0, 1]$ the discount factor to differentiate the value of rewards now vs those received in the future. In finite horizon cases $\gamma = 1$ and can be ignored.

The learner then has the objective of funding a policy function from the set of all policy functions $\Pi : \mathcal{S} \to \mathcal{A}$ which solves the maximisation of the expected discounted sum of rewards:

$$\pi^* = \underset{\pi \in \Pi}{\operatorname{argmax}} E\Big[\sum_{t=0} \gamma^t R(s_t, a_t)|\pi\Big]$$

The policy function is often a probability distribution over actions $\pi(a|s) = \mathbb{P}(a|s)$ $\forall a \in \mathcal{A}$, $s \in \mathcal{S}$. I will restrict attention to deterministic strategies.

As previously mentioned, the Markovian property of this process comes from the transition function. It is satisfied if the probability of transition to a new state is determined only by the current state and chosen action.

The value function of a policy π, written $V^\pi : \mathcal{S} \to \mathbb{R}$ is the expected discounted sum of rewards from following policy π when in any state s. Similarly the State Action Value function $Q^\pi : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is the expected discounted sum of rewards from following policy π after choosing any action a in any state s

7.3.3 Shields for safe learning

A shield (Alshiekh et al., 2017; Jansen et al., 2020; Pranger et al., 2020) is a structure developed in the field of Safe Reinforcement Learning (RL) literature which is able to prevent the agent from breaking certain behavioural laws. It is a filtering function that tells the agent what actions are legal at any moment in time. The agent then makes their selection from the choice of legal actions, acts and the environment responds with a new state and reward as with standard RL.

Often the presence of a shield is unacknowledged because the environment automatically prevents the agent from making impossible actions (like moving eastwards when already positioned in the northeast boundary of a maze). This can be seen in Sutton and Barto (1998) where the Action space is in places written as a function of the state.

The shield effectively restricts policy exploration to the subspace of legal policies which form a subset of the full policy set. Alternative approaches to Safe RL avoid illegal behaviour by passing penalties for bad behaviour directly through the reward function thereby altering the optimization criterion (García and Fernandez, 2015). This has drawbacks because it will often require penalty functions to be differentiable, and the penalties do not necessarily avoid 'illegal' behaviour in training. A related approach is to apply constraints to the policy (such as minimum reward requirement, but possibly a restriction over the entire state trajectory) so the problem becomes one of constrained optimisation (often a constrained MDP).

A positive feature of the shield approach to constrained learning is that it does not interfere with the mechanics of the learning process other than consuming information and is largely agnostic to the technique employed. This is attractive from the perspective of the designer as different learning algorithms can be tried at little incremental cost. It also has positive privacy implications because it allows the possible use of a shield in situations where the workings of the Agent are not known to shield owner.

Shields are present during learning, but their presence might not be necessary in deployment. In theory the learned policy function will not include any weighting on illegal policies, and so should be safe left unsupervised. However novel situations could be encountered on deployment, so the shield can be left in place to ensure illegal actions are never selected. This comes at the computational cost of having to consult the shield before taking every decision.

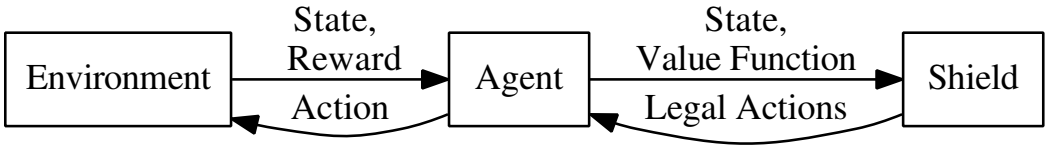


Figure 7.1: The Shield structure in Reinforcement Learning

In the application presented here, the shield needs access to the agent's state action value function so as to assess whether, at the point of order placement, the agent would be disappointed on immediate execution. This additional information requirement is atypical for shields and too our knowledge, novel. As Ashton (2020, 2021c) points out, a shield structure for enforcing arbitrary laws is likely to require information about the agent's policy and a model of the environment as well as the history of states and actions. This is to allow assessment of agent causality and intent to see if they meet the actus reus and mens rea requirements of any specific law. In the case of spoofing, the action part of the offence is only reliant on the conduct of the agent at single point in time (the placement of orders)[8].

[8]In practice, Spoofing and the related practice of Layering might involve the placement

Definition 7.1 can be interpreted with the value function V^π introduced at the beginning of this section. Remember the value $V(s)^\pi$ of being in any state s is the expected discounted sum of rewards from following policy π. The expected reward from order placement (joining a queue) according to strategy π is simply $V^\pi(s)$ when $\pi(s) \in \{join^j\}_{i=1...N}$. This means when the policy function suggests an order should be placed, the value function can be checked against a benchmark value $b(s)$ which corresponds to the best legitimately expected outcome at that state. Equivalently the Value-Action function $Q^\pi(s,a)$ is the expected cumulative reward for choosing action a in state s and following π thereafter. $Q^\pi(s,a)$ can be checked for all actions $a \in \{join^j\}_{i=1...N}$ against the benchmark reward to check when a queue can legitimately joined.

The process the shield undertakes is shown in 7. It uses a simple labelling function for convenience which indicates whether the agent is in queue or not, defined by:

$$L(s) = \begin{cases} queue^j \text{ if } y^j > 0 \\ out \text{ if } y^j = 0 \, \forall j \in 1,..,N \end{cases} \tag{7.4}$$

In the setting I have described, the benchmark variable corresponds to everyone in the queue being served immediately and is invariant when the agent is in queue $b(s') = r^{small}$ for $L(s') \neq out$ and otherwise undefined.

7.3.4 Experiment Method

A Reinforcement Learning Agent was put into the environment as discussed in Section 7.3.1. The learning algorithm chosen was Proximal Policy Optimisation (PPO) which is type of Actor Critic method. PPO was introduced by Schulman et al. (2017) and extended by Huang and Ontañón (2020) to cover the case when actions are masked from the agent when they are not allowed

of multiple orders, at different price levels possibly at different venues.

Algorithm 7: Spoofing Intent Shield

Data: s_t, State Action function $Q(a, s)$, Reward function $R(s, a, s')$, Labelling function L, benchmark value b and $S^{out} := \{s \in \mathcal{S} | L(s) = out\} \subseteq \mathcal{S}$
Result: A set of admissible actions
if $s_t \notin S^{out}$ **then**
  $legal\_actions = \emptyset$;
  for *each join choice* $a^j \in \mathcal{A}$ **do**
    if $Q(s_t, a^j) \leq b$ **then**
      $legal\_actions \mathrel{+}= a^j$;
    end
  end
else
  $legal\_actions = \mathcal{A}$;
end

according to some exogenous mechanism.

PPO is an evolution of A3C (Mnih et al., 2016b) and TRPO (Trust Region Policy Optimisation) (Schulman et al., 2017) in the sense that it uses multiple workers to evaluate policies and that policy updates each period are limited in order to limit the problems associated with catastrophic forgetting (Goodfellow et al., 2015; Kemker et al., 2018). This is an observed phenomena where RL agents would suddenly learn policies which were much worse than previous iterations and has been observed feature of neural networks learning tasks sequentially since the 1980s McCloskey and Cohen (1989).

The StableBaselines3 package for python (Raffin et al., 2021) has an experimental repository named *SB3-Contrib* which contains an implementation of Maskable PPO. I adapted this to implement the shield. I kept the learning coefficients at their default values including the design of the policy and value networks (shared 2 hidden layers of 64 neurons with tanh activation). The size of the state and action space were not particularly big, nor the complexity of the problem particularly high, so hyper parameter tuning was not necessary for the task.

The action space of the agent was discrete, equalling the number of queues (2)

plus a wait or do nothing action.

The state space of the agent was a vector of length 13, containing the following variables:

- *Q_Emptied* - Binary variable, length $N = 2$ - The queue emptied
- *Exit* - Binary variable - Agent paid and exited.
- *EWOP* (Exit with out paying) - Binary - Agent exited without paying because cashier left.
- *In_Queue* - Binary - Whether Agent in any queue or not.
- *Position_in_queue* - Non-negative integer, length $N = 2$ - The Agent's position in every queue.
- *Length_of_queue_* - Non-negative integer, length $N = 2$ - The length of every queue.
- *Arrivals* - Non-negative integer, length $N = 2$ - The number of arrivals in every queue that period.
- *Departures* - Non-negative integer, length $N = 2$ - The number of departures in every queue that period.

Using two queues in the experiment simplified the shield considerably. When in a queue, the agent could wait, move to the back of the queue that were in and move to the back of the other queue. Since moving to the back of their own queue is clearly not an optimal strategy (except when spoofing), this was ruled out by default using the shield. The shield could then concentrate on comparing the value of swapping queues (if recommended) with the benchmark. If the value was higher than the benchmark level, the action was changed to wait by default.

The strategies were all evaluated using a vectorised version of the environment for at least 10,000 trials. A trial lasted at most 200 periods in training but was unconstrained in evaluation. Each environment instance was reset when the agent exited a queue and a new starting state for that instance was drawn

from an initial state buffer. This was to avoid episodes being serially correlated (a queue that had just emptied would otherwise have a smaller than average queue length next period).

7.3.5 Parameters

As previously mentioned, the number of queues N was set to 2.

The reward for reaching the front of the queue and being served was $r_{low} = 1$ and the reward for leaving the queue without paying was $r_{high} = 5$. In each period, a small loss of -0.025 was incurred. This serves a dual purpose as a waiting cost and a method to improve learning, since all actions have a non-zero reward. The agent had a discount rate γ of 0.99.

The Customer arrival function followed a memoryless Poisson process $W^j \sim P(\alpha^i)$ with $\alpha = [2, 2]$ if the queue length at the beginning of time t represented by $x^j_{t_1}$ was less than or equal to 8, otherwise the queue would not receive any arrivals for that period.

The departure function of customers (those served by the cashier) followed a Poisson process $V^j \sim P(\delta^i)$ with $\delta = [1, 1]$ with the restriction that V^j was limited by the length of the queue post arrivals so negative queues could not occur.

The random queue emptying process $F^{qmax} \sim B(\phi_t^{qmax})$ was Bernoulli distributed for the longest queue at the beginning of period t:
$qmax_t = \text{argmax}_j(x^j_{t-1})$ with $\phi_t^{qmax} = 0.001 x^{qmax}_{t-1}$. That is to say the probability of the longest queue (or Queue 1 in the event of a tie) completely emptying at the beginning of the period was proportional to the length of that queue at the end of the previous period.

By making the probability of queue emptying linear with queue length, the average length of queue was limited. This would not otherwise be the case

given that the arrival rate was greater than the departure rate. The limit of 8 on the arrival function served several functions. It compressed the distribution of queue lengths which aided the learning process by making the state space smaller. It also decreased the average wait time which increased the relative attractiveness of waiting in the queue versus lingering in the hope that the queue would empty. By making the emptying process only apply to the longer queue at any moment in time (or queue 1 in a tie), an additional feature about the environment needed to be learned, thereby penalising random strategies which proved surprisingly efficient in this environment under certain parameterisations.

7.4 Results

The learned strategies were benchmarked against the following set-strategies:

1. **Shortest** The Agent joins the shortest queue in every period.
2. **Shortest Once** The Agent join the shortest queue initially and then waits.
3. **Any Once** The Agent joins any queue initially and then waits.
4. **One Action** The Agent keeps (re)joining queue one.
5. **Two Action** The Agent keeps (re)joining queue two.
6. **Random** The Agent chooses actions at random
7. **Spoof**: The Agent (re)joins the longest queue in every period.
8. **Cutoff(n)**: The agent follows the *'Shortest'* strategy whilst queue length is less than n, otherwise follow *'Spoof'*.

Two types of learned strategies were trained.

1. **Benchmark Shield** Of the type described by Algorithm 7. The Agent is always restricted from joining back of queue it is already in, and can only join another queue if the expected value of doing so is below some threshold. This was chosen to be 1, which was the reward for reaching the front of the queue and being served (r_w), this could be thought of as

the best possible *legitimate* outcome in any period.

2. **No Shield** The agent was free to learn any strategy.

Strategy	Periods	Discounted Reward		Duration		Reward		Reward Type	
		Mean	Std	Mean	Std	Mean	Std	1	5
Full Shield	10,005	0.9169	0.9703	10.134	4.1971	1.0045	1.0106	93.55%	6.45%
No Shield	10,003	1.0088	2.067	81.459	77.9694	1.8755	2.647	27.20%	72.80%
Shortest	10,020	0.8758	0.814	9.1138	3.5642	0.9526	0.849	95.49%	4.51%
Shortest once	10,006	0.879	0.8435	9.3582	3.6355	0.9575	0.8764	95.21%	4.79%
Any_queue_once	10,005	0.8879	0.9311	10.2577	3.8909	0.9738	0.9629	94.24%	5.76%
One action	10,001	0.5121	2.231	135.4879	136.6426	1.1508	3.642	11.55%	88.45%
Two action	10,000	-0.1051	2.1736	187.9782	188.9272	-0.3467	4.937	16.18%	83.82%
Random	10,004	0.6166	2.0805	96.6813	95.8421	1.2343	3.042	33.72%	66.28%
Spoof	10,000	1.4341	2.1331	90.3947	88.768	2.7373	2.222	0.07%	99.93%
Cutoff	10,020	0.9467	1.045	10.6354	5.3165	1.0411	1.0954	92.33%	7.67%

Table 7.1: Strategy Comparison Results. The grey rows correspond to hard-programmed benchmark strategies, the two white rows are learned strategies.

Table 7.1 shows the mean discounted reward, average duration and reward type mix from the three learned strategies and the benchmark strategies. The mix of rewards allows us to understand the goal of the strategy to some extent. The *Spoof* strategy for example, achieves practically all (99.9%) of its returns with rewards which occur when the queue emptied rather than rewards from reaching the front of the queue. This means that it is able to achieve the highest mean discounted reward of 1.434. At the other extreme, *Shortest* strategy achieves 95.49% of its rewards from reaching the front of the queue and only 4.51% from the queue emptying.

Of the learned strategies, the *No Shield* strategy achieves the highest discounted reward of 1.009 over an average duration of 81.459 periods. 72.8% of its rewards occur from queue emptying. This level is above the 66.28% proportion achieved by the *Random* strategy. The discounted average reward is higher (1.009 vs 0.617 for random) and the average duration is lower (81.459 versus 96.681 periods for random). The *Full Shield* strategy achieves a mean discounted reward of 0.917 with an average duration of 10.134 periods. These results ware close to the three standard queuing benchmark strategies *Shortest* (Disc mean 0.876, duration 9.114), *Shortest Once* (Disc mean 0.879, duration 9.358 and *Any Queue Once*(Disc mean 0.888, duration 10.258). 93.55% of the

rewards in the learned *Full Shield* strategy occur from reaching the front of the queue. This is close to the conventional queuing strategies of 95.49% for Shortest, 95.21% for *Shortest once* and 94.24% for *Any queue once*. Note that discounted reward for *One action* is higher than *Two action*. The reward for always choosing action 1 is higher than always choosing action 2 since queue 1 is more likely to empty because in the situation where the queues are the same length, only queue 1 is eligible to empty.

Strategy	**Action type**			**Not in Queue**	**Rejoin rate**	**Swap rate**	**Wait rate**
	Wait 0	**Join 1**	**Join 2**				
Full Shield	87.21%	10.76%	2.03%	1.77%	0.00%	2.98%	94.88%
No Shield	16.29%	42.01%	41.69%	3.55%	11.28%	68.70%	15.94%
Shortest	84.90%	8.37%	6.73%	0.00%	0.00%	4.71%	95.29%
Shortest_once	89.31%	6.50%	4.19%	0.00%	0.00%	0.00%	100.00%
Any queue once	90.25%	4.89%	4.86%	0.00%	0.00%	0.00%	100.00%
One action		100.00%		0.00%	100.00%	0.00%	0.00%
Two action			100.00%	0.00%	100.00%	0.00%	0.00%
Random	33.32%	33.34%	33.34%	1.71%	32.20%	32.46%	33.32%
Spoof		58.21%	41.79%	0.00%	78.13%	21.87%	0.00%
Cutoff	66.88%	20.43%	12.70%	0.00%	14.53%	5.43%	80.04%

Table 7.2: Strategy Analysis. The grey rows correspond to hard-programmed benchmark strategies, the two white rows are learned strategies. Strategies that rejoin more are more likely to benefit from queue emptying.

Table 7.2 attempts to analyse the characteristics of the strategies. A number of different statistics were extracted from the results by converting the sequence of actions and states into a regular language and then using regular expressions (regexes) to identify behaviour of interest. This method was used to calculate the number of times the agent rejoined their own queue, swapped queues, waited in queue, normalised by the length of time the agent spent in queue. The number of periods the agent waited outside a queue was normalised by the total length of the episode[9]. Looking at the mix of actions chosen we can see that those strategies which are associated with spoofing (*No Shield*, *One action*, *two action* and *Spoof*) do not wait as much as the standard queue and wait strategies of *Full Shield*, *Shortest*, *Shortest once* and *Any queue once*. Waiting is the only action which leads unambiguously to promotion and being

[9]Typically the episode length and time spent in queue had a difference of 1 since most strategies spent no time outside a queue

served when already in a queue. The general higher frequency of Action 1 reflects the fact that the queue is more likely to empty (since it is chosen in the event of equal queues).

Table 7.2 shows that the two learned strategies (Full Shield and No Shield) both spend 1.77% and 3.55% of periods not in a queue. The queue's dynamics are memoryless (or Markovian) and so any time not spent in a queue is wasted. The learned strategies have therefore not converged to a locally efficient state. The rejoin rate in Table 7.2 corresponds to the proportion of periods that the agent, when in a queue, joins the back of a queue they are already in. This is an unambiguously bad strategy for reaching the front of a queue, but pretty good[10] for waiting for a queue empty event. This can be seen by the high occurrence of this action type in the strategies that achieve a higher proportion of the $r_{high} = 5$ rewards (*Spoof, One Action*, *Two Action*). The *Full shield* strategy never rejoins the same queue through the shield's operation. In contrast the No Shield strategy has a rejoin rate of 11%.

The swap rate in Table 7.2 corresponds to the proportion of periods that the agent, when in a queue, joins the back of another queue. If the agent joins a shorter queue this reduces their expected time, but if they join a longer queue, then it increases their expected wait time and increases their probability of leaving a queue without paying. Unlike the rejoin rate, the swap rate is not an unambiguous sign that a strategy is targeting a particular outcome. We can see that the *Shortest* strategy swaps queues 4.71% of the time spent in queue. This occurs when the other queue becomes shorter than the one it is in. The *Full Shield* is similarly constrained in its swap rate at 2.98%. The unconstrained *No Shield* strategy has a swap rate of 68.70% which indicates a targeting of the higher reward associated with queue emptying since this is much higher than the rate at which the *Shortest* strategy swaps. The *Spoof* strategy swaps queues 21.87% of the time; this indicates that the *No Shield*

[10]It might be even better to join the back of the other queue if it is longer

strategy swaps more than it should (and rejoins less than it should).

Strategy	Queue Joins				In Queue Swaps			
	#	Shorter	Longer	Same	#	Shorter	Longer	Same
Full Shield	10,005	46.37%	32.93%	20.70%	2,967	74.08%	17.29%	8.63%
No Shield	10,003	3.96%	74.41%	21.63%	579,713	49.66%	27.99%	22.34%
Shortest	10,020	79.56%	0.00%	20.44%	3,766	100.00%	0.00%	0.00%
Shortest once	10,006	80.11%	0.00%	19.89%	-			
Any queue once	10,005	40.92%	39.64%	19.44%	-			
One action	10,001	41.00%	39.45%	19.56%	-			
Two_action	10,000	39.88%	40.51%	19.61%	-			
Random	10,004	40.08%	39.53%	20.38%	317,317	47.71%	32.41%	19.87%
Spoof	10,000	0.00%	80.34%	19.66%	197,639	0.00%	74.26%	25.74%
Cutoff	10,020	79.12%	0.00%	20.88%	6,720	0.00%	72.80%	27.20%

Table 7.3: Strategy Rationality Analysis. The grey rows correspond to hard-programmed benchmark strategies, the two white rows are learned strategies.

Table 7.3 shows some rough rationality measures for the strategies. The first measure considers whether on joining a queue initially, whether the shorter or longer queue is joined. At two extremes, the *Shortest* and *Shortest Once* strategies never join the longer queue whilst the *Spoof* strategy will always join the longest queue. Of the learned strategies, the unrestricted strategy learns to join the longest queue (joining the shorter queue 3.96% of the time). The *Full shield* strategy favours joining a shorter queue but still joins the longer one 32.93% of the time. The second measure in table 7.3 considers queue swaps (to a different queue). Once again, at the extremes, the *Shortest* strategy will only ever swap queue if it is shorter whilst the *Spoof* Strategy never swaps for a shorter queue. The *Full shield* strategy has learned to favour swaps to a shorter queue, only joining longer queues 17.29% of the time and the same length queue 8.63% of the time. The unrestricted *No shield* strategy joins longer queues 27.99% of the time and the same length queue 22.34% of the time. These figures are close to the *Random* strategy's swap figures so the *No shield* strategy arguably has further learning to do. One additional feature to notice is that the *No shield* strategy is far more active swapping than any other strategy as testified by the number of swaps (approx $580,000$). This figure is dependent on mean duration, so not easily comparable between

strategies, nevertheless the strategy's mean duration is not the highest (81.459 periods versus 96.681 for the *Random* strategy), indicating queue swapping is a key feature of the learned unconstrained strategy.

In Q	In Q 1	In Q 2	Original Action	Corrected Action	Count	Proportion
No	No	No	Wait outside		1,760	
No	No	No	Join 1		9,608	
No	No	No	Join 2		397	
Yes	No	Yes	Wait in 2		3,346	3.73%
Yes	No	Yes	Swap to 1	Wait in 2	264	0.30%
Yes	No	Yes	Rejoin 2	Wait in 2	28	0.03%
Yes	No	Yes	Swap to 1		1,306	1.46%
Yes	No	Yes	Rejoin 2	Swap to 1		0.000%
Yes	Yes	No	Wait in 1		80,678	90.02%
Yes	Yes	No	Rejoin 1	Wait in 1	2,330	2.60%
Yes	Yes	No	Swap to 2	Wait in 1	11	0.01%
Yes	Yes	No	Swap to 2		1,661	1.85%

Table 7.4: Shield Intervention Analysis for *Full shield* strategy. The highlighted rows correspond to active intervention by the shield.

The agent was evaluated with the shield in place, but to measure the necessity of that shield, the percentage of decisions overruled by the shield was counted ('disagreement'). 2.94% of decisions were overruled by the shield. This could be considered a convergence measure related to training since with a sufficiently long training period, the agent's policy would eventually converge to a policy allowed by the shield and disagreement would be zero. The precise occurrences of when the shield overruled the policy function are shown in Table 7.4, with the disagreement cases highlighted in grey. The states where the agent is not in a queue are not considered here because the implemented shield did not place any restrictions on the agent when they weren't in a queue. The single largest case where the shield overrules the policy function is when the agent is in queue 1 and wants to join the back of queue 1. This is a behaviour which unambiguously does not further the agent's cause in reaching the front of a queue and supports the position that the learned strategy is not fully converged.

In summary the results show that the shield does prevent the agent from

queuing behaviour which avoids trying to reach the front of the queue. Without the shield, the learned behaviour exhibits features of a strategy which attempts to benefit from queue emptying events.

7.5 Discussion

The type of RL algorithm which I use (PPO) produces a policy network which produces a recommendation of which action to choose depending on the state and a value network which is the agent's estimate of future discounted reward following their policy in any given state. In the case when the policy function advised swapping queues, the shield's job was to check the value function against the threshold. The threshold value was chosen to represent the best possible legitimate result from joining a queue - reaching the front and being served immediately. If the agent's estimated value of swapping to another queue was higher, then the shield would prevent the queue swap. This was because this type of action indicates that the aim of the policy was not to terminate the episode by reaching the front of the queue but instead loitering and increase the risk of benefiting from a queue emptying occurrence. The cost of this calculation is one forward pass through the value network per period.

Note that if more queues were present, then this approach would not work, since the value function only evaluates the expected value from following the given policy. In this case an Action Value function (aka Q function) is needed, that is to say a function that estimates the value of being in a certain state, choosing a certain action and then following a policy thereafter. The shield would then need to make as many forward passes as there are alternative queues in order to screen out any invalid swaps. In practice these passes would be performed as a vectorised operation and would not add a significant computation burden to the shield algorithm. Actor critic methods are perfectly able to estimate Q-functions instead of Value functions but Value functions are used in typical implementations. Alternatively, a Q-function can be derived

from an Agent's Value function through a one-step simulation, from the starting state, choosing the action to be evaluated, then taking the expectation of the ensuing reward and value function of the following state. The advantage of this approach is that it does not force the agent to produce an Q function, allowing standard actor critic implementations to be used, the disadvantage being the time it takes to form an expectation over ensuing states and their values and the requirement that the shield has access to a simulation of the environment. If the (simulated) environment has been programmed in such a way as to accept array inputs, the additional computational time might not be too expensive, compared with just being given the agent's Q function, since the agent's value network which is a neural network also naturally accept array inputs.

The validity of the method presented here comes from the ability to understand the best-case reward of joining a queue ex-ante, and knowing that expected rewards in excess of this imply that the agent's policy is trying to take advantage of non-legitimate reward sources. This is just as it would be in a limit order book where the best possible result - immediate order lifting with minimum market impact, is known at the point of order placement. This feature of the spoofing problem will not generalise to many other intent problems. Typically, we will not know what the agent's future policy is trying to achieve just by examining their value function. Additionally, a model-free RL agent has no possible means of predicting what the next state of the world will be, and so they themselves cannot know what they will do or not do in the future unless the magnitude of the estimated rewards informs them of some feature of their policy as is the case here.

The CFTC's guidance CFTC (2013) on the interpretation of spoofing regulations suggests a non-exhaustive list of alternative motivations to spoof other than price movement. These include denial of service type attacks designed to overload an exchange, submission of orders to delay someone else's execution,

submission and cancellation to create a false impression of market liquidity. Other motivations might exist; for example a trader who was testing a market to understand the likely reaction to a future planned order was prosecuted for spoofing CFTC (2018b). An obvious difference with the environment used in this experiment and actual markets is the absence of a price analogue. The wording of the spoofing prohibition and the subsequent guidance makes it clear that ignoring market impact is not a critical problem when diagnosing spoofing and supports our method to some extent here.

The task of producing a sensible benchmark value is made harder when we consider trading strategies over a longer period of time which are composed of many individual order placements. If we assume trades are filled instantly in the best-case scenario to achieve the best case reward which forms the benchmark, then we must also consider wider market supply and demand dynamics. Assuming very high demand or supply will result in artificially high benchmarks which the learning algorithm is able to undershoot with a profitable spoofing strategy. That is why in the method presented, I consider atomic decisions which translate to the decision to place atomic orders. The estimation of demand and supply curves is not an unstudied task in trading applications and is information that can be included in deciding the benchmark value in situations where it is not suitable to analyse atomic decisions.

Some methods of model-free RL do not involve estimating either a Value or Action-Value function and therefore the method presented here based on disappointment inferred intent will not work. Policy gradient methods directly estimate a differentiable parameterised policy to produce a probability distribution over all actions conditioned on state. Actor-Critic methods such as PPO and A2C Mnih et al. (2016b) sit in between Policy gradient and Value based methods and produce a value function in addition to the policy function. These methods are efficient because many parts of the process to produce Policy and Value gradients are shared. Any policy gradient could therefore be

altered in the design stage to additionally calculate a value function, even if it were not used in the learning process directly and only an input to the shield. In the worst-case, a value function can be recovered from a policy function and a simulation of the environment by the shield.

Model-based RL algorithms seek to either to learn a model of the world or are given one, and then proceed to learn a policy using some sort of planning technique. Most famously, Monte-Carlo tree search was used by Alpha Zero (Silver et al., 2017) to look for winning Go strategies. More recently, an algorithm has been developed to combine the planning techniques of AlphaZero whilst learning its own model of the environment that it finds itself in Schrittwieser et al. (2020). In both cases a value function is estimated and used to avoid exhaustive policy search. RL agents with models of the world are able to assess the probability that they will cancel an order at the point of placement. It is tempting to conclude that this information is sufficient for intent assessment, but this is wrong because it does not answer the 'why' an order placement decision has been placed. Traders placing orders at the end of the trading session might know that there is very little chance that the order will be lifted, and yet they still do so because they hope that it will. A control mechanism that prevented orders being placed that were almost certainly likely to be cancelled might have undesirable effects on liquidity. In the very extreme case where the algorithm foresaw the probability of order cancellation in all possible cases, we might conclude that it was engaged in a spoofing strategy. I believe that an analysis of the aim of actions is unavoidable when assessing intent.

7.6 Summary

In this chapter I look at the problem of controlling the 'intent' of a Reinforcement Learning (RL) trained trading agent. In particular, I look at intent to cancel an order at the point of its placement which is the SEC definition of the market manipulative practice of spoofing. I consider a statistical measure

for intent but reject it in favour of a counterfactual account. This considers a caused outcome (order cancellation) to be intended if the agent is disappointed if it does not occur (the order is executed). This is informed by legal reasoning and the definition of intent found in the US Model Penal Code.

In the second part of the chapter, I introduce a minimal queuing environment which captures three key features of the spoofing problem in limit order books; order placement and cancellation have analogous actions, the trader is not in complete control of the fate of an order and the reward structure is such that avoiding order execution can be rewarding. I then present structure called a shield which screens out 'unsafe' actions available to the RL agent at each time period. This structure is otherwise generally unaware and agnostic to the particular learning algorithm being used by the agent provided a value function is made available to it. To our knowledge this is the first example of a shield being used on value functions. Here 'unsafe' is taken to mean those actions which have a higher estimated value than the one which would occur from leaving the queue immediately. This ensures the agent cannot be disappointed about immediate execution, they therefore cannot intend to avoid order execution.

The results of training an RL agent using PPO (Proximal Policy Optimisation) in the environment both with and without the shield show that unconstrained the agent would learn a 'spoofing' type strategy and learn a legitimate strategy focused on fast execution when constrained by the shield. Various statistics concerning the behaviour exhibited in each strategy support the two strategies finding different approaches to the problem. Most obviously this can be seen by the mix of rewards received by each strategy, the unconstrained strategy receiving a high proportion of rewards received from queue emptying episodes. A variety of benchmark strategies including an idealised spoofing and efficient queuing policy are used to provide a meaningful frame of reference against the learned strategies. They show that whilst the unconstrained and constrained

strategies share characteristics with the ideal spoofing and fastest queuing strategy respectively, they do not perfectly match them. This coupled with analysis concerning the rationality of certain decisions made by the learned strategies indicate that learning in both cases has not fully converged to efficient queuing policies.

In the discussion of the method and the experiment I discuss how the unusual reward characteristics of a limit order book allow the shield to constrain intent by reference to the agent's value function. In this case, intent can be construed and controlled without a model of the environment. Construction of a meaningful benchmark figure with which to assess disappointment would require some knowledge and assumptions about market conditions, nevertheless the techniques presented in this chapter could be in spoofing prevention and detection applications.

Whilst value-based RL algorithms are popular, not every algorithm produces one. A value-based shield could still be constructed by estimating the value function derived by the policy in conjunction with a model of the environment and a set of rationality and efficiency constraints on the likely agent reward function. This would be an application of Inverse Reinforcement Learning where behaviour streams are used to construct rewards functions.

Chapter 8

Conclusions and Future work

This chapter concludes the thesis, summarises the content of chapters 4-8, identifies their contributions to research and suggests some future research directions.

8.1 Summary

This thesis takes a simple research question rooted in computer science and finance - "Do trading algorithms learn to manipulate a limit order book by spoofing and how do we stop that from happening". It proceeds along a path from multi agent simulations with deep reinforcement learning through Law and Psychology before returning to computer science with a plausible definition which can be used to control the intent of an algorithm. Whilst the primary subject matter concerns the regulation of algorithms in the marketplace, the conclusions on the subject of intent in AI and people's perceptions about that are applicable to any use of AI which requires a legally informed concept of intent.

Interdisciplinary research requires additional effort and often involves moving out of personal comfort zones. However, it is necessary when conducting work on the regulation of AI. Computer scientists do not make market regulations

and they will not be asked to interpret the law in the event of an algorithm committing some AI crime. Equally, as any number of recent scandals illustrate, for example Kramer et al. (2014); Verma (2014); González-Esteban y Patrici Calvo (2022); De Cremer and Kasparov (2021), engineers alone should not be trusted to decide what is reasonable or not when considering the behaviour of algorithms. Indeed Responsible Innovation is an avowed objective of the EPSRC (Owen, 2014).

I will now briefly summarise the findings of each chapter in the thesis and any future extensions or research directions for the work.

8.1.1 BUCLSE Platform

For a system to be manipulable it needs to be changeable. Likewise, to assess whether an agent can manipulate an environment, the agent must be able to change the environment in the first place. This motivates the building of what has recently been termed 'a digital twin', aka an interactive environment which responds to the actions of the agents within it.

The study of market manipulation requires such an environment, and this chapter presents the design of an interactive Limit order book simulator programmed in Python.

8.1.1.1 Future extensions

Python is an attractive language because it is generally accessible and readable. However, this incurs computational overheads and as a result its performance is slow outside specially optimised software packages.

Deep Reinforcement Learning in its basic form is data intensive. This coupled with the aforementioned speed issues of a market environment programmed in Python makes research slow.

An improvement to the simulator platform would be to rewrite parts of it in a faster language. It is being refactored into Rust. Another solution is to vectorise the behaviour and performance of the various different agents in the simulation. This approach was taken in the environment featured in Chapter 7. Both approaches do come with a cost of interpretability which should be born in mind.

Computer science is best progressed by improving existing packages not least because many eyes have already gone some way to debugging them. Progress with BUCLSE should be stopped until the suitability of ABIDES (Byrd et al., 2019) for market manipulation research can be assessed.

8.1.2 Emergence of spoofing in a RL LOB environment

This chapter shows that even a limited instance of a Reinforcement Learner will learn a strategy which is equivalent to spoofing when deployed in a LOB environment and given a simple state representation of the market.

The LOB is populated with a variety of Zero intelligence traders, Heuristic Belief Traders taken from Wang and Wellman (2017) and a class of trader which explicitly uses order imbalance to make buy or sell decisions.

A problem with the use of neural networks to create policy functions is that they are difficult to interpret. I resolve the problem by fitting a decision tree to the policy by observing the inputs and outputs of the neural network. Evidence of spoofing is seen by the change in the orderbook imbalance statistic and the improved profitability that agents with the ability to spoof have. The emergence of spoofing behaviour in a simulator might just mean that the RL agent learns how to game the environment that they train in and on deployment they fail to do so. This is a specific example of the kind of reward gaming seen in Amodei et al. (2016) and should offer a warning to people trying to train trading algorithms in this way. I also note that spoofing does not require

success, so a bad spoofing algorithm is just as culpable for the owner as a good one.

8.1.2.1 Future extensions

The shield device and the definition of intent defined in Chapter 7 could be adapted to verify that spoofing is taking place in this environment. This would not be onerous given that I used Q-learning which produces a state-action value function.

It would be interesting to test the finding with a less restrictive action space and termination conditions. For instance, the liquidation problem in Cartea et al. (2020), where an agent has to sell a multiple unit position but is allowed to hold more units on the path to zero.

A multi-agent reinforcement experiment would be interesting to see whether a spoofing strategy is stable in equilibrium or whether agents learn not to trust order book information as a reliable indicator of supply and demand.

8.1.3 A definition of intent for algorithms

The definition of spoofing in the USA is the placement of orders with the intent to cancel them before execution. This presents an obvious problem when considering what constitutes spoofing behaviour in an algorithm which has learned its behaviour. The creator of the algorithm might not have intended for it to spoof.

Spoofing is not the only crime to rely on the intentional status of an action to qualify as such. Indeed, nearly all crimes require some culpable sort of mental state (mens rea) to accompany the prohibited action (actus reus). Typically, the degree of mens rea informs us about the culpability of the harm caused by the actus reus. However, some crimes, such as spoofing and more generally those involving a degree of deceit require mens rea in the actus reus. Here

actions are only deemed criminal if they are performed with some sort of intent or purpose at the point of commission. The presence of these crimes justifies defining intent irrespective of the harder debate to be had about the moral status of wrongs caused by algorithmic actors.

In this chapter I review how the common law defines intent and attempt to translate this into a logical construct which is amenable to being translated into code.

The endeavour is complicated by the lack of a unified definition of intent and the existence of different degrees of intent (Direct, Oblique, Recklessness and Negligence) as well as temporal modes - inchoate or ulterior intent.

We present definitions of all of these intent types in language specific enough to be applied to AI but general enough to be agnostic about its particular design.

8.1.3.1 Future extensions

The definitions could be translated into a more formal language, some variant of temporal logic (Kleinberg and Mishra, 2009) which would be able to express notions of causality and desire as I suggest in Ashton (2021c). This would then perhaps allow formal verification methods to be applied to check behaviour for legality (Baier and Katoen, 2008).

8.1.4 Testing a definition of intent for algorithms on laypeople

This experiment was written and conducted with Matija Franklin and Professor David Lagnado, members of the causal cognition research group at UCL which is within Experimental Psychology. It asks the twin research questions of whether a provided definition of intent coincided with a folk-definition and whether lay-people judged the intent of an AI any differently from that of a Human.

The experimental setting was of a parcel delivering unmanned drone flying through a zone as quickly as possible whilst avoiding dangerous no-fly zones.

Over the course of three separate but related experiments, we find very little difference in people's perceptions of intent between human and AI actors. The simple definition of direct intent is also effective at recreating the folk concept of intent embedded in the test subjects.

8.1.4.1 Future extensions

Vignette studies are often shown to be dependent on the setting they describe. It would be interesting to test our findings in more settings with different types of AI actor to see how robust they are.

8.1.5 A method to identify and restrict intent to cancel in a simple queuing example

In this chapter I use the findings of Chapter 5 to craft a definition of intent specific to spoofing. Firstly I show that any statistical definition of intent based on the probability of cancellation at order placement cannot definitively prove intent. Instead, I present a definition based on counterfactual reasoning and regret. I argue that a trader intends to cancel an order at the point of placement if they would be disappointed should the order actually be executed. This definition includes cases where the motivation of spoofing is not to make money but something else. It is therefore in line with the CTFC definition and guidance.

I use a minimal example of a queuing game to present a possible technique that would allow safe algorithmic trader training without the risk of spoofing. In the game, the agent needs to choose a checkout queue in a shop in order to pay for an object and leave. They would like to choose the fastest moving queue. Occasionally, the queue will empty, and the members of that queue can leave

without paying. This eventuality has a higher reward than the one where the agent has to pay for the object. The agent can either decide to join a queue where they will be served quickly, or they can loiter by switching queues and attempt to wait for this queue emptying scenario. This game captures elements of the spoofing problem without the overhead of a multi-agent simulation.

I present a RL training method that does not learn a loitering (spoofing) policy by using a shield structure. At the point of any action being chosen, the shield interrogates the value function of the learner to check the expected reward for that action. The shield prevents actions with expected rewards higher than one occurring from immediate serving (order execution). This is possible because the reward for immediate serving is known ex-ante, just as it is for order-execution on a LOB. Without the shield in place, the agent is shown to learn a loitering policy.

The definition and shield are as far as I know, the only attempt in research to define intent to cancel and the only method to restrict the intent of a reinforcement learning agent.

8.2 Contributions

In this thesis I have shown that the risk of an auto-didactic algorithm learning a market manipulative called spoofing is foreseeable. This was achieved by building a fully functioning limit order book environment and populating it with a variety of zero intelligence type trading agents. Foreseeability is an important attribute when considering the law because it is a benchmark required for negligence and recklessness. In the process of demonstrating the spoofing behaviour of the RL trained algorithm, I used a number of different techniques which might be used to interpret the behaviour of a black box trading algorithm.

The second half of the thesis considers the problem of identifying and control-

ling intent in an algorithm. Firstly, I present a definition of direct, means-end, ulterior and oblique intent informed by existing legal theory on the subject of mens rea. Aside from spoofing, many fraud related crimes require a working definition of intent in order to be defined. Fraud is an offence that algorithms could easily be accused of in the future given their growing use in chat and e-commerce and finance applications. Without a definition of intent suitable for algorithms, certain offences could not be proven against algorithmic actors, thereby opening a responsibility gap and an opportunity for bad actors.

Courts in the UK have juries comprised of laypeople who are routinely asked to judge the intentional state of the accused when presented with evidence. At some point the intent of an algorithmic actor will need to be assessed by a jury. Courts will need to provide instruction as to what intent might mean for an algorithmic actor and be confident that the jury is willing and capable of ascribing intent to an algorithm. Chapter 6 explores these issues and finds that a given definition of intent correlates with people's natural understanding of the term and that the differences between ascriptions of intent are small when contrasting between human and AI actors.

Finally, I present a well reasoned definition of intent, based on legal theory, which is suitable for assessing intent to cancel. It can be used by prosecutors and algorithm owners to test for spoofing but more broadly any application where algorithmic intent needs to be measured. Using this definition, I show how a shield can be used in the training of a reinforcement learning agent to control for intent. This is to my knowledge the first solution to intent-control in reinforcement learning and has applications in the legal control of algorithms which can be viewed as an application of AI alignment Russell (2019). It is also the first quantitative experiment on the subject of spoofing that explicitly tackles its intentional nature.

Appendices

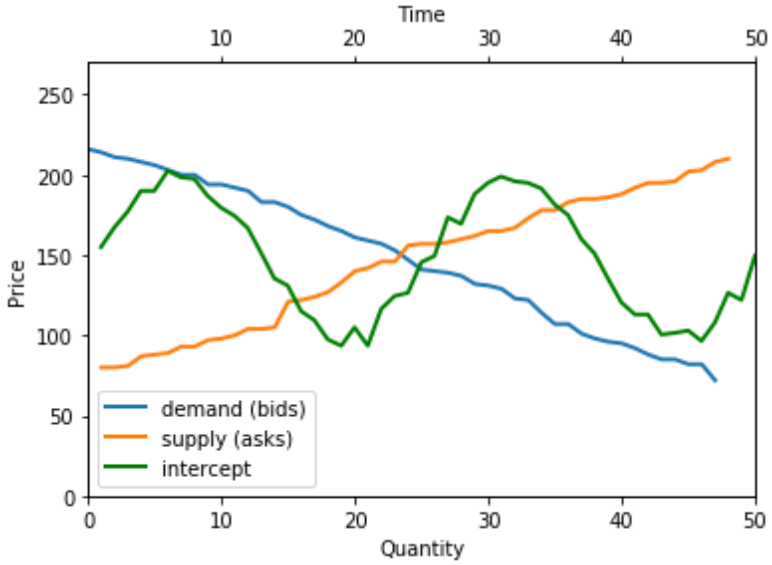


Figure A.1: Equilibrium price defined over time, as defined by the intersection of demand and supply curves at any moment in time. In this figure, the demand and supply curves are taken as of t=50, hence their intersection is in line with green curve at t=50.

A BUCLSE - Supply and Demand setup

In this setup, there is a list of buy and sell 'client' orders defined over time which can be in turn used to define supply and demand curves. Order prices are distributed over some user defined, time varying range. From this order list a curve can be calculated by plotting cumulative quantity of orders below any price for demand and cumulative over any price for supply. In other words, the curves define the quantity theoretically in the market at any period of time that is willing to sell above a certain price or buy below a certain price. Demand declines as quantity increases whilst supply increases as price increases. Classic microeconomics tells us that equilibrium is found where the two lines meet; where demand equals supply. Typically the demand and supply curves are linear at any moment in time, but they can be translated according to a user defined function, thereby giving a shifting equilibrium over time, see figure A.1.

The BSE setup gives us freedom to define how the curves are formed, how they

change over time and at what rate the orders are distributed to traders. One experimental improvement from the original BSE formulation is that BUCLSE pre-specifies the orders and their destination before the experiment begins (originally this is done on demand). We are therefore able to rerun the same experiment if required (albeit with potentially different stochastic behaviour from the traders). This aided with debugging and allowed greater flexibility over fixing a seed for the random number generators involved.

A.1 Trader Bestiary - Supply Demand type

These traders are taken from the BSE experiment setup where there is a Supply and Demand object which provides traders with 'client' orders randomly throughout the experiment. These client orders are either buys or sells with a limit price, and the traders are tasked with executing at these limit prices or better. Traders do not hold inventory and execute only in a single direction at any time. These trader types all appear in Cliff (2018).

A.1.1 GiveAway

When prompted, trader submits order for price equal to limit price of client order.

A.1.2 ZIC

Trader submits orders for prices randomly selected between client order limit price and a some percentage of the limit price. This type of zero intelligence trader appeared in Gode and Sunder (1993).

A.1.3 Shaver

Trader improves best bid (ask) by a price increment if their client limit order price is higher (lower) than best bid (ask). As in Cliff (2018).

A.1.4 ZIP

First appearing in Cliff and Bruten (1997), traders maintain a randomly initialised, buying margin $m_b \in (-1, 0]$ and selling margin $m_s \in [0, \infty)$ which represent the minimum profit margin they require to transact at. When invited to submit orders, the limit price is calculated as $\lambda(1 + m_i)$ for $i \in \{b, s\}$ where λ is the client limit price for the order. The restriction on m_s and m_b mean that the trader will always submit orders no more or no less than the client's limit depending on whether it was a buy or sell. Associated with the client limit λ and profit margin is a target price $\bar{p}$. Following every trade on exchange, the trader updates their target price and then margin as a function of target price, depending on the circumstances of that trade. In the case of the trader having a sell client order, they will react as follows:

- If the last submission to the exchange resulted in a transaction at price p_{last} then:
  - If p_{last} is greater than the traders target price $\bar{p}$, then the trader increases $\bar{p}$ with respect to p_{last}, and their margin m_s is adjusted accordingly.
  - Else if the ask queue contracted (an ask order is 'lifted' by a buyer) at a price lower than $\bar{p}$ and the trader has an order to offload the trader will lower their target price $\bar{p}$ with respect to p_{last} and margin accordingly m_s
- Else there was no trade
  - If best ask decreases (is improved), and target price $\bar{p}$ is greater than best ask, trader chooses a price better than best bid.

The case of buy order is analogous.

For a reference price p, the increase in target price $\bar{p}$ is calculated by first

calculating an intermediate value $\tilde{p}$ as follows:

$$\tilde{p} = p * relative\_shift + absolute\_shift = p(1 + 0.05 * \epsilon) + 0.05 * \psi \quad \text{for } \epsilon, \psi \sim U[0,1] \quad (1)$$

and the decrease analogously:

$$\tilde{p} = p * relative\_shift - absolute\_shift = p(1 - 0.05 * \epsilon) - 0.05 * \psi \quad \text{for } \epsilon, \psi \sim U[0,1] \quad (2)$$

The updated margin m' and target price $\bar{p}'$ are then calculated with the following iteration:

$$\Delta' := \tilde{p} - \bar{p} \quad (3)$$

$$C_t := (1 - momentum)\beta\Delta' + momentum * C_{t-1}\Delta \quad (4)$$

$$m_i' := \begin{cases} \frac{\hat{p}+C_t}{\lambda} - 1 & \text{if } > 0 \text{ for seller, and } < 0 \text{ for buyer} \\ m_i & \text{else} \end{cases} \quad (5)$$

$$\bar{p}' := \lambda * (m_i' + 1) \quad (6)$$

$$\Delta = \Delta' \quad (7)$$

for trader specific parameters $momentum \sim U[0, 0.1]$ and $\beta \sim 0.1 + U[0, 0.4]$ defined on instantiation.

A.2 Market session setup: Supply and Demand

The sequence of events that occurs during a period differs between the Supply and Demand and Fundamental Price type configurations of the environment. For supply and demand, it is as follows:

On initiation, the order and trader submission sequence is defined. The sequence for a period then proceeds as follows:

1. New customer orders are dispatched to traders
2. A trader is picked to submit an order to the exchange
3. Any resulting trades are processed
4. Traders update their records according to changes in the LOB.

B Reinforcement Learning

B.1 Relationship of action value function Q and state value function V

The Value Function associated with policy π at state s is defined as:

$$V_\pi(s) = E_\pi[R_t|s_t = s] = E_\pi\Big[\sum_{k=0}^{\infty}\gamma^k r_{t+k+1}|s_t = s\Big] = E_\pi\Big[r_{t+1} + \gamma\sum_{k=0}^{\infty}\gamma^k r_{t+k+2}|s_t = s\Big]$$
$$= \sum_a \pi(s,a)\sum_{s'} P^a_{s,s'}\left[R^a_{s,s'} + \gamma E_\pi\Big[\sum_{k=0}^{\infty}\gamma^k r_{t+k+2}|s_{t+1} = s'\Big]\right]$$
$$= \sum_a \pi(s,a)\sum_{s'} P^a_{s,s'}[R^a_{s,s'} + \gamma V_\pi(s')] \quad (8)$$

The last line is known as the Bellman equations for V_π

The optimal, unique value function $V^*(s)$ is defined as follows:

$$V^*(s) = \max_\pi V_\pi(s) \quad (9)$$

Equation 8 is modified to form the Bellman optimality equations for V^*:

$$V^*(s) = \max_a \sum_{s'} P^a_{s,s'}[R^a_{s,s'} + \gamma V^*(s')] \quad (10)$$

Recalling that the optimal Action Value (Q) Function Q^* is defined as:

$$Q^*(s,a) = \max_\pi Q_\pi(s,a) \quad \text{for all } s,a$$

The two concepts can be combined thus:

$$Q^*(s,a) = E[r_{t+1} + \gamma V_*(s_{t+1}) | s_t = s, a_t = a] \quad \text{and} \quad V^*(s) = \max_a Q^*(s,a)$$

This follows since the definition of $Q_\pi(s,a)$ is the value of taking action a in state s and taking policy π thereafter. Thus $Q^*(s,a)$ is the value of taking action a in state s and taking the optimal policy π^* thereafter.

B.2 Gradient descent for Q-learning

The use of gradient descent in Q-learning is motivated by the desire to solve the following, for a distribution of states and actions P(s,a):

$$\boldsymbol{\theta}^* = arg\min_\theta \sum_{S,A} P(s,a)\Big(Q^*(s,a) - Q(s,a,\boldsymbol{\theta})\Big) \tag{11}$$

P weights the importance of errors in different state action pairs. One choice of P would be the distribution of rewards and actions that the RL agent encounters when using the optimal strategy π^*.

Assuming that we can sample from this distribution, then the gradient descent derived update to find θ^* is:

$$\begin{aligned}\boldsymbol{\theta}_{t+1} &= \boldsymbol{\theta}_t - \frac{1}{2}\alpha(t)\nabla_{\boldsymbol{\theta}_t}\Big(Q^*(s_t,a_t) - Q(s_t,a_t,\boldsymbol{\theta_t})\Big) \\ &= \boldsymbol{\theta}_t + \alpha(t)\nabla_{\boldsymbol{\theta}_t}\Big(Q^*(s_t,a_t) - Q(s,a,\boldsymbol{\theta}_t)\Big) \\ &= \boldsymbol{\theta}_t + \alpha(t)\nabla_{\boldsymbol{\theta}_t}\Big(E[r_{t+1} + \gamma \max_a Q^*(s_{t+1},a)] - Q(s,a,\boldsymbol{\theta}_t)\Big)\end{aligned} \tag{12}$$

Unfortunately we do not know $Q^*(s_{t+1},a)$ (since that is our ultimate objec-

tive). We substitute it with our best estimate which is $Q(s_{t+1}, a, \boldsymbol{\theta}_t)$, and the expectation operator is tackled because we are sampling data. This leads us to the update:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t + \alpha(t)\Big(Y_t^Q - Q(s_t, a_t, \boldsymbol{\theta}_t)\Delta_{\theta_t} Q(s_t, a_t, \boldsymbol{\theta}_t)\Big) \tag{13}$$

Where $\alpha(t)$ is a declining update size and Y_t^Q is the known as the target function and defined as:

$$Y_t^Q := R_{t+1} + \gamma \max_a Q(s_{t+1}, a, \boldsymbol{\theta}_t) \tag{14}$$

B.3 Double Q

The Q learning update:

$$Q_{t+1}(s_t, a_t) = Q_t(s_t, a_t) + \alpha_t(r_t + \gamma \max_a Q_t(s_{t+1}, a) - Q_t(s_t, a_t)) \tag{15}$$

Converges to the optimal value function Q^*, which solves the following equation:

$$\forall s, a : Q^*(s, a) = \sum_{s'} P_{s,a}^{s'}(R_{s,a}^{s'} + \gamma \max_a Q^*(s', a)) \tag{16}$$

where $P_{s,a}^{s'}$ is the probability of ending in state s' from state s and choosing action a, similarly for reward $R_{s,a}^{s'}$.

The Q learning update can be thought of a problem of estimating the value of the next state using the following approximations:

$$\max_a Q_t(s_{t+1}, a) \approx E[\max_a Q_t(s_{t+1}, a)] \approx \max_a E[Q_t(s_{t+1}, a)] \tag{17}$$

However the order of the expectation operator and the max operator is not in general swappable. Whilst the first term is an unbiased estimator of the second, it is not the third.

In Van Hasselt (2010), the author shows that by maintaining two Q function approximators built with mutually exclusive data, an unbiased estimator for $\max_a E[Q_t(s_{t+1}, a)]$ can be obtained.

Theorem Appendix.1. Van Hasselt (2010) Let $X = \{X_1, \ldots, X_m\}$ be a set of R.V. and let $\mu^A := \{\mu_1^A, \ldots \mu_M^A\}$, $\mu^B := \{\mu_1^B, \ldots \mu_M^B\}$ be two sets of unbiased estimators of X s.t. $E[\mu_i^A] = E[\mu_i^B] = E[X_i]\ \forall i$.

Define $M := \{j\ | E[X_j] = max_i E[x_i]\}$. Similarly define $a^* = \text{argmax}_i\ \mu_i^A$. Then

$$E[\mu_{a^*}^B] = E[X_{a^*}] \leq \max_i E[X_i] \tag{18}$$

The inequality is strict iff $P(a^* \notin M) > 0$

Proof Assuming $a^* \in M$, then $E[\mu_{a^*}^B] = E[X_{a^*}] := \max_i E[X_i]$. Else $a* \notin M$, then $E[\mu_{a^*}^B] = E[X_{a^*}] < E[X_j] := \max_i E[X_i]$ for some j

Then since a^* is in M or not we must have:

$$\begin{aligned} E[\mu_{a^*}^B] &= P(a^* \in M) E[\mu_{a^*} | a^* \in M] + P(a^* \notin M) E[\mu_{a^*} | a^* \notin M] \\ &= P(a^* \in M) \max_i E[X_i] + P(a^* \notin M) E[\mu_{a^*} | a^* \notin M] \\ &\leq P(a^* \in M) \max_i E[X_i] + P(a^* \notin M) \max_i E[X_i] = \max_i E[X_i] \end{aligned}$$

The authors therefore propose an alteration to the update in 15. Assuming estimators Q^A and Q^B and defining $a^* := \text{argmax}_a\, Q^A(s', a)$ we have:

$$Q_{t+1}^A(s, a) = Q_t^A(s, a) + \alpha_t(r_t + \gamma Q_t^B(s', a^*) - Q_t^A(s, a)) \tag{19}$$

Similarly b^* is defined w.r.t Q^B. Updates of Q^B and Q^A are performed alternately. Data efficiency should not be effected because a can be chosen using both Q^B and Q^A

B.4 Deterministic and stochastic policies

The optimality of deterministic policies, those $\Pi_d \subset \Pi : A \times S :\rightarrow [0,1]$ such that $\forall s \in S$, $\exists\ a \in A$ s.t. $\pi(a,s)) = 1$ is closely related to non-deterministic (or stochastic) policies.

Consider a problem of the following form for some objective function R:

$$\max_d R(d) \quad \text{for} \ \ d \in D : S \rightarrow A. \tag{20}$$

Then any solution of equation 20 must also solve the following maximisation over all distributions of D

$$\max_{p(d)} E_p[R(d))] \quad \text{where } p(d) \in P : S \times A \rightarrow [0,1] \tag{21}$$

if d^* solves equation 20 than the indicator probability function over d^* gives the same maximal value. In expectation no distribution over policies can be better this fixed policy hence maximising over stochastic policies is equivalent to maximising over fixed policies.

B.5 Q-learning: Full update

The Q-learning update samples one possible proceeding state s' and reward r from to update the value initial state action pair (s,a). This updating of state action values using a single sample can result in large sample errors. This is could be a problem in our trading application where reward distributions exhibit high variance with respect to the action chosen.

In a process termed 'full backup' in Sutton and Barto (2018), Q-value iteration will update the value of (s, a) using the full distribution of rewards and subsequent states (r, s') using the conditional probability distribution $P(r, s'|s, a)$. This is not possible unless an estimate of the probability distribution is maintained or provided. Since the tabular Dyna-Q setup maintains an empirical model of the environment $\hat{P}$ (possible since the experiment has a finite number of proceeding states from any initial state), it is a natural extension to perform full backups. The Q function is then updated as follows:

$$Q^*_{new}(s, a) = \sum_{s' \in \mathbb{S}} \hat{P}(s'|s, a)\Big(r(s, a, s') + \gamma \min_b Q^*_{old}(s', b)\Big) \tag{22}$$

Kalathil et al. (2014) term the method 'Empirical Q-Value Iteration', provide some convergence properties, rates and comparisons versus the sample backup method typically used. They update using a sample of states, thus extending the method to situations where the state action space is infinite.

B.6 Variational Autoencoders (VAEs)

Variational Auto Encoders (VAEs) Kingma and Ba (2014), Rezende et al. (2014) come from Bayesian belief networks. They were motivated by the desire to generate novel samples from an arbitrary, simple distribution given some training data X without the need for time consuming Makov Chain Monte carlo sampling.

The intuition behind the VAE is not straightforward, and requires a more technical explanation which we have adapted from Doersch (2016). The process aims to sample z in a lower dimension latent variable space using a known distribution family (typically this will be multivariate Gaussian) which we'll call $P(Z)$. The process then fits a neural network f parameters θ such that $x \sim \mathcal{N}(f(z, \theta), \sigma.I) := P(X|\, z, \theta)$. This is possible since any n dimensional distribution can be estimated by the composition of any 'simpler' n dimensional

distribution and a sufficiently complicated function f. We can find θ through:

$$\theta^* = argmax_\theta P(X_{train}) = \int_z P(X_{train}|\ z,\theta).P(z).dz \tag{23}$$

Maximising this equation will take an unfeasibly large amount of training data so VAE attempts to only sample z in latent space which are likely to have produced X_{train}. In other words an encoding function $Q(z|X,\psi)$ with parameters ψ is introduced to allow the feasible sampling of $E_{z\sim QP(X_{train}|z)}$.

To get a good Q we minimise KL divergence of $Q_\psi(z|X)$ and $P(z|X)$, which working through the definition of KL divergence and the application of Bayes rule on $P(z|X)$ to reverse the order of conditioning, we get the fundamental equation:

$$logP(X) - KL[Q_\psi(\tilde{X}|Z)||P(z|\tilde{X})] = E_\psi z \sim Q[logP_\theta(\tilde{X}|z)] - KL[Q_\psi(z|\tilde{X})||P(z)] \tag{24}$$

Where $\tilde{X} = X_{train}$.

Noting that KL divergence is greater than zero, the right hand side of equation 24 gives us a lower bound on $logP(X)$, commonly called the evidence lower bound (Barber, 2016). Since $Q_\psi(z|X)$ maps variables from feature space to latent space it is the encoder, and $P_\theta(X|z)$ is the decoder.

In practice $Q_\psi(z|X)$ is typically chosen to have multivariate gaussian distribution $N(z|\mu(X,\psi_\mu),\Sigma(X,\psi_\Sigma))$ where as before the mean and variance functions are formed from a deep neural network parametrised by ψ. This means that the KL divergence has an analytical form (because it is between two Gaussian distributions). Researcher have come to recognise this is a limitation of the VAE method because there are limited distribution pairs with this property Creswell and Bharath (2017) and Gaussian latent space

The right hand side of equation 24 can be maximised using standard stochastic gradient descent after a slight 'reparametrisation trick' (Kingma and Welling (2013)) to allow differentiability when using stochastic gradient descent to train the model. This is shown in fig B.1: Instead of sampling from $N(z|\mu(X,\psi_\mu),\Sigma(X,\psi_\Sigma))$ we sample from z from $\mu(X,\psi_\mu)+\Sigma(X,\psi_\Sigma).\epsilon$ where $\epsilon \sim \mathcal{N}(0,I)$.

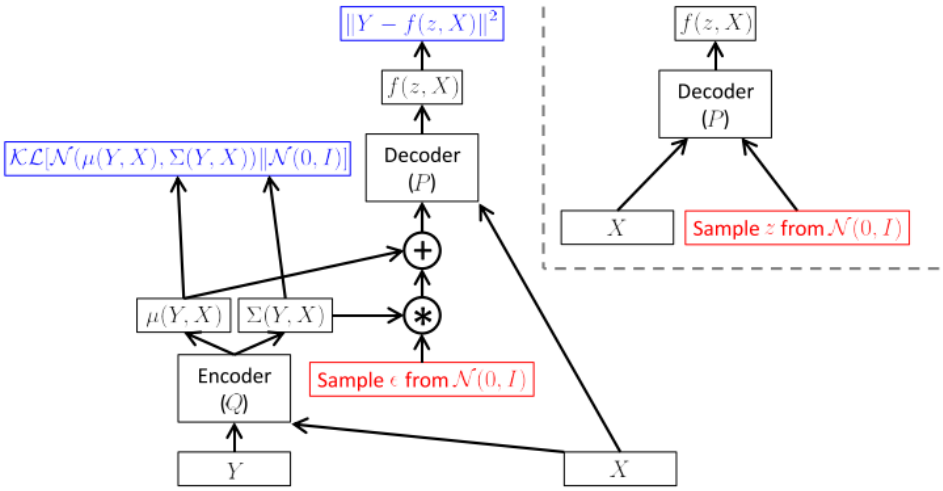


Figure B.1: Training for the VAE requires affine parameterisation of latent z to allow differentiability in training

Conditional VAEs (Walker et al. (2016)) and (Sohn et al. (2015)) exist which allow the VAE framework to tackle 'gap filling' problems (Doersch (2016)). Applied to images, the CVAE can fill in missing, structured, multimodal data (typically images) Sohn et al. (2015) or predict future video frames (Walker et al. (2016)). The mathematics behind the CVAE are the same to the VAE except the encoding and decoding networks are conditioned on some input variable (Doersch (2016)). By conditioning on an input, the user excludes that information from being encoded in latent space since it is available to the encoder and decoder. Conditioning allows the user to choose the type of output they want. Repeated sampling will produce a distribution which should match the observed empirical distribution. In Chapter 4 a prediction network f_ψ is trained to predict the state transition of the environment. Estimation of

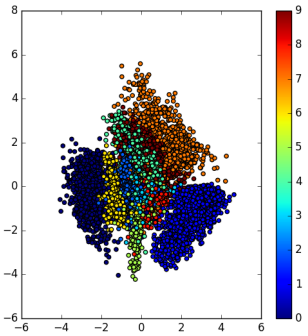


Figure B.2: VAE clusters different figures of the MNIST dataset in a 2d latent space

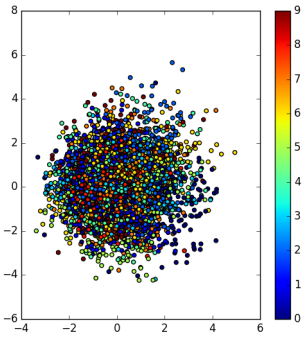


Figure B.3: CVAE does not cluster different figures in latent space because it has been conditioned on them, instead each figure should be gaussian distributed

The inference model Q is a probabilistic encoder in the sense that for a given x, it finds the parameters of the distribution that z is drawn from. By extension this means that the reconstruction of x will be a distribution not a single point and thus anomaly detection can be measured using a more interpretable measure - ie probability of reconstruction $E_\psi z \sim Q[logP_\theta(\tilde{X}|z)]$ which is the approach of An and Cho (2015).

f_ψ is achieved through the framework of Conditional Variational Autoencoders (CVAEs). In the training process an encoder $g_\alpha : S \times S \times A \to Z$, parameterised by $\alpha \in \mathbb{R}^n$ is trained back to back with the decoder, where composition of the encoder and decoder thus forms a something akin to a autoencoder: $h := f \circ g : S \times S \times A \to S$. The loss function for the composed neural networks is:

$$\mathcal{L}_{AE} := \alpha||S' - h(S,(S',A)))||_{RECON} + \beta\mathcal{KL}(g(S',(S,A),\mathcal{N}(0,1))) \quad \alpha,\beta \in \mathbb{R} \tag{25}$$

Where $||\circ||_{RECON}$ is a distance metric, (typically l_2), thus making the first term reconstruction loss. The second term $KL(\circ,\circ)$ is the KL divergence between the encoding and the user defined distribution of Z which is chosen to be normal for algebraic convenience. I have enclosed the term (S,A) to emphasise that these are the conditioning variables in the process.

A realisation of the decoder for prior state s and action a is then produced by

drawing z from $\mathcal{N}(0, I^d)$. A diagram of the training and sampling procedure for the CVAE can be seen in figure B.1

B.7 UCB

The problem of balancing exploration and exploitation has been examined in the literature surrounding multi-arm bandit problems. Reinforcement Learning research has been slow to adopt the advances in this area, preferring to stick to simple ϵ-greedy exploration strategies which are demonstrably inefficient (at least from a regret perspective). In this section we will quickly introduce some concepts in multi-arm bandit search and prove that a strategy known as UCB, is efficient in some sense.

B.7.1 Statement of problem

For time periods $t = 1, \dots T$ a learner can choose an action $a \in \mathbb{A} = \{a_1, a_2, \dots a_k\}$.

Nature chooses some payoff r_a conditional on a from distribution $\pi_a := P(r|a)$.

Define cumulative regret:

$$R_T := E\Big[\sum_{t=1}^{T}(U^* - U_t)\Big] \quad \text{for } U^* := \max_a E[r|a] \quad \text{and } U_t = E[r|a_t] \tag{26}$$

R_T can be rewritten equivalently as as count of each action taken and the ensuing reward:

$$R_T = \sum_a E[N_t(a)]\Delta_a \quad \text{for } N_t(a) := \sum_{t=1}^{T}\mathbb{1}(a = a_t) \quad \text{and } \Delta_a := U^* - U_a \tag{27}$$

$N_t(a)$ is the number of selection for action a made during a history $H =$

$(a_1, a_2, \dots a_T)$ and Δ_a is the gap between an optimal action and the actual action taken.

A benchmark for a successful strategy is to aim for a regret term which is not linear in time, that is to say $O(R) \leq T$.

A greedy strategy (one of exploitation) would be to play the action with the highest current estimated payoff. Greedy strategies have a high chance of locking onto a suboptimal action implying that regret is not going to be sublinear.
Proposition B.1. ϵ-greedy strategy has linear regret

Proof. Consider the common ϵ-greedy strategy which selects a random action uniformly with probability ϵ and otherwise chooses the strategy with the best current estimated payoff. This strategy will also have linear regret since even if the best action is found, the exploration choices will occur ϵT of the time and have non zero regret. □

B.7.2 Optimism in the face of uncertainty

Suppose the agent imagines all plausible environments compatible with their experience. They then select the most favourable action based on this. By plausible we could think about confidence bounds[1] or interquartile range around the estimate of the reward from choosing action a. For any action a, its reward estimate becomes more accurate as it is chosen more. In this way we can see that suboptimal actions will only be chosen for a limited time before others are explored.

Define the empirical reward of action a to be:

$$\hat{U}_t(a) := \frac{1}{N_t(a)} \sum_{t=1}^{T} r_t \mathbb{1}(a_t = a) \tag{28}$$

[1] UCB=Upper Confidence Bound

where $\mathbb{1}(\cdot)$ is the usual indicator function.

The following proposition and proof is adapted from Krause (2009).

Proposition B.2. If actions are chosen according to:

$$a_t = \underset{a}{\operatorname{argmax}} \hat{U}_t(a) + \sqrt{\frac{log(t)}{N_t(a)}} \tag{29}$$

And each action is chosen at least once ($N_t(a) \geq 1 \quad \forall a$)

Then the regret is sub-linear in the limit of T, and in particular:

$$E[R_T] = O(\frac{K \log T}{\Delta}) \tag{30}$$

where $\Delta = \min_a \Delta_a$ and some constant K.

From equation 29 it can be seen that the confidence bound term $\sqrt{\frac{log(t)}{N_t(a)}}$ grows as more actions are tried, but decreases the number of times any single action is chosen. Actions are tried infinitely often, but exploration and exploitation are balanced.

Proof. Proof proceeds to bound the expectation of $E[N_t(a)]$ for all actions a.

Let there be a the bound $B_t(a)$ on any action such that $P(U(a) < \hat{U}_t(a) + B_t(a))$ is high. Strategy is then to choose action $a_t = \operatorname{argmax}_a \hat{U}_t(a) + B_t(a))$

The Hoeffding bound states that for i.i.d. random variables $X_1, \ldots X_t$ in $[0,1]$ such that $E[X] = \mu$ and their sample mean $\tilde{X}_t = t^{-1} \sum_{\tau=1}^{t} X_\tau$ then for some constant c: $P[\tilde{X} > \mu + c] \leq e^{-tc^2}$

Thus $P[U(a) > \tilde{U}_T(a) + B_T(a)] \leq e^{-2N_T(a)B_T(a)^2}$.

Choosing $B_T(a) = \sqrt{\frac{\log T}{N_T(a)}}$ means that:

$$P[U(a) > \tilde{U}_T(a) + B_T(a)] \leq t^{-2} \tag{31}$$

Suppose at time t<T, a suboptimal action a is chosen. Thus:

$$\hat{U}_T(a) + B_T(a) \geq \hat{U}_T^* + B^*(T) \tag{32}$$

adding $(\mu_j - \mu_j)$ to the LHS and deducting μ^* from both sides:

$$\underbrace{\hat{U}_T(a) - (\mu_j + B_T(a))}_{A} + \underbrace{(\mu_a - \mu^* + 2B_T(a))}_{B} \geq \underbrace{\hat{U}_T^* - (\mu^* - B^*(T))}_{-C}$$

At least one of A,B,C must be greater than or equal to zero. This gives three inequalities respectively:

$$\hat{U}_t(a) \geq \mu_a + B_t(a) \tag{33}$$

$$\mu_a + 2B_t(a) \leq \mu^* \tag{34}$$

$$\hat{U}_t^* \leq \mu^* - B^*(t) \tag{35}$$

From equation 31 we know that the first and third inequalities happen with probability less than T^{-2}.

If $N_t(a) \geq \frac{4log(T)}{\Delta_a^2} := l$ then

$$\mu_a + 2B_T(a) = \mu_a + 2\sqrt{\frac{2log(T)}{N_T(a)}} \leq \mu_a + 2\sqrt{\frac{2log(T)}{8log(n)}\Delta_a} = \mu_a + \Delta_a = \mu^*$$

Finally we are able to bound $E[N_t(a)]$ by decomposing the expectation into two exclusive conditional ones:

$$E[N_t(a)] =$$
$$\underbrace{P(N_T(a) \leq l)}_{\leq 1} \underbrace{E[N_T(a)|N_T(a) \leq l]}_{\leq l} + \underbrace{P(N_T(a) > l)}_{\leq 2T^{-4}} \underbrace{E[N_T(a)|N_T(a) > l]}_{\leq T}$$
$$\leq l + 2T^{-2} \quad (36)$$

We know $P(N_t(a) > l) \leq 2T^{-2}$ since this would violate either condition 33 or 35. The union bound is then used with the knowledge either condition only happens with probability at most T^{-2}.

From the definition of regret:

$$R_t = \sum_a E[N_t(a)]\Delta_a \leq \sum_a (l + 2T^{-1})\Delta_a = \sum_a \frac{4log(T)}{\Delta_a^2}.\Delta_a + 2T^{-1}\Delta_a$$

From which we can conclude:

$$O(R_T) = O\Big(\frac{Klog(T)}{\Delta}\Big)$$

□

A similar but longer proof exists where T the number of plays is not known in advance. In this case, the second term in the UCB strategy can become $B_t(a) = \sqrt{\frac{2\log t}{N_t(a)}}$ and the limit is only minimally changed to $O\Big(\frac{K\log t}{\Delta}\Big)$. Related limits also exist on the number of times a su-boptimal action can be chosen. Interestingly, the regret bound can be shown to be at least logarithmic in steps, so UCB is efficient up to constants. See Auer and Ortner (2010) for more details.

B.7.3 UCB in Q-learning

Using UCB style exploration instead of ϵ-greedy has come relatively recently to Reinforcement literature. Jin et al. (2018) show that using UCB with Q-learning gives some performance guarantees on regret which have hitherto been missing. These move the efficiency of Q-learning on a par with Reinforcement Learning with a model.

B.8 Actor Critic Methods

Actor Critic methods differ from Q-learning in that they have a representation of the policy directly rather than deriving it in a greedy fashion from the Q-function. In the case of deep learning this means that a neural network parameterised by θ estimates the policy function $\pi_\theta : \mathcal{S} \rightarrow \mathcal{A}$ for some set of parameters θ. This is the Actor network which is learned in addition to the parameterised Action-value (Q) or more commonly the Value function $V^\pi : \mathcal{S} \rightarrow \mathbb{R}$.

The method is based in policy-based learning where gradient ascent is attempted on $E_{\pi_\theta}[R_t] := E_{\pi_\theta}[\sum_{k=0}^{\infty} \gamma^k r_{t+k}]$. Various different updates of the parameterisation are possible, one introduced by Williams (1992) is known as REINFORCE and updates θ in direction $\nabla_\theta log\pi(a_t|s_t,\theta)R_t$ which can be shown to be an unbiased estimator of $\nabla_\theta E[R_t]$. In practice, lower variance can be achieved by deducting a baseline from the reward R_t so the update becomes $\nabla_\theta log\pi(a_t|s_t,\theta)(R_t - V^\pi(s_t)$. The sign of the final term in brackets indicates how surprising the reward was from taking action a_t in that state.

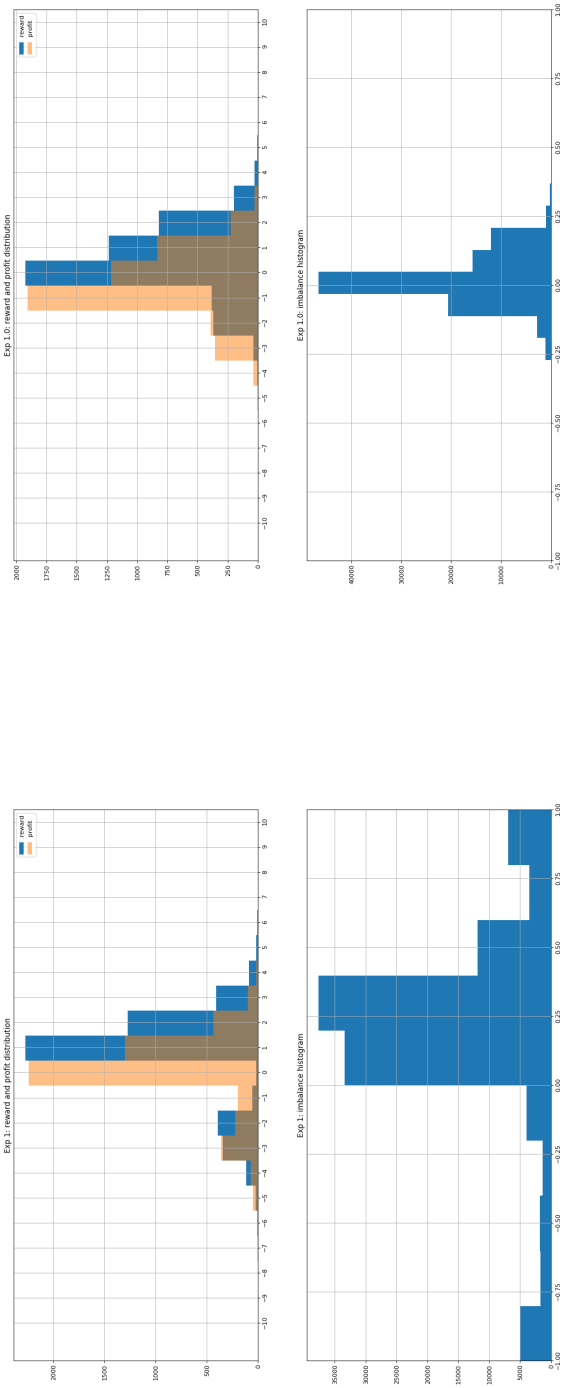
Of course the value function is not known, so it is also estimated by a neural network parameterised by ψ: $V^\pi_\psi(s_t) \approx V^\pi(s_t)$. Gradient descent is performed to minimise the difference between the parameterised value function and the actual value function for that policy: $J(\psi) := E_\pi[((V^\pi(s) - V^\pi_\psi(s))^2]$. Parameters ψ are updated in the direction $(V^\pi(s_t) - V^\pi_\psi(s_t))\nabla_\psi V^\pi_\psi(s_t)$. The role of

the value function estimate in the update of the actor network can be thought of as a critic. For more details see Mnih et al. (2016b).

C Chapter 4: Supplementary Results

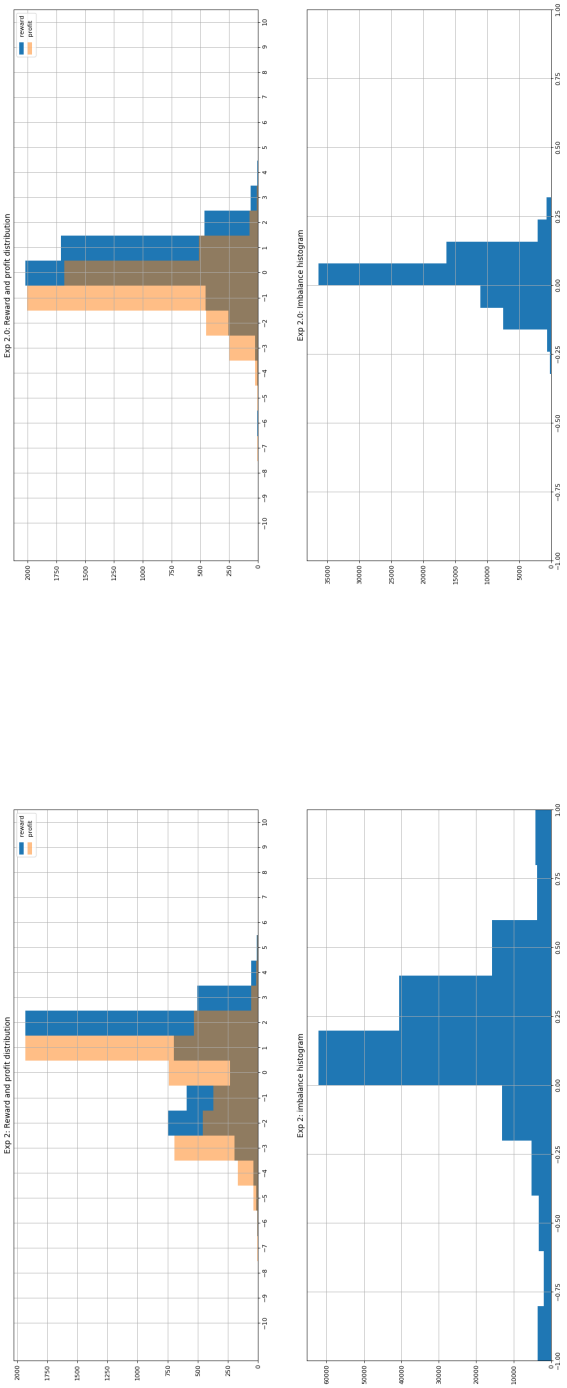
C.1 Chapter 4: Reward and Imbalance distributions

The charts in the following pages show the reward distributions of the strategies and the distributions of orderbook imbalance. In the experiments where order placement is possible, and this option is utilised, the distribution is distributed across its domain. Where spoofing is successful, I would expect the distribution to be more skewed to the positive. The left hand side of the distribution is affected by the agent cancelling their large orders.



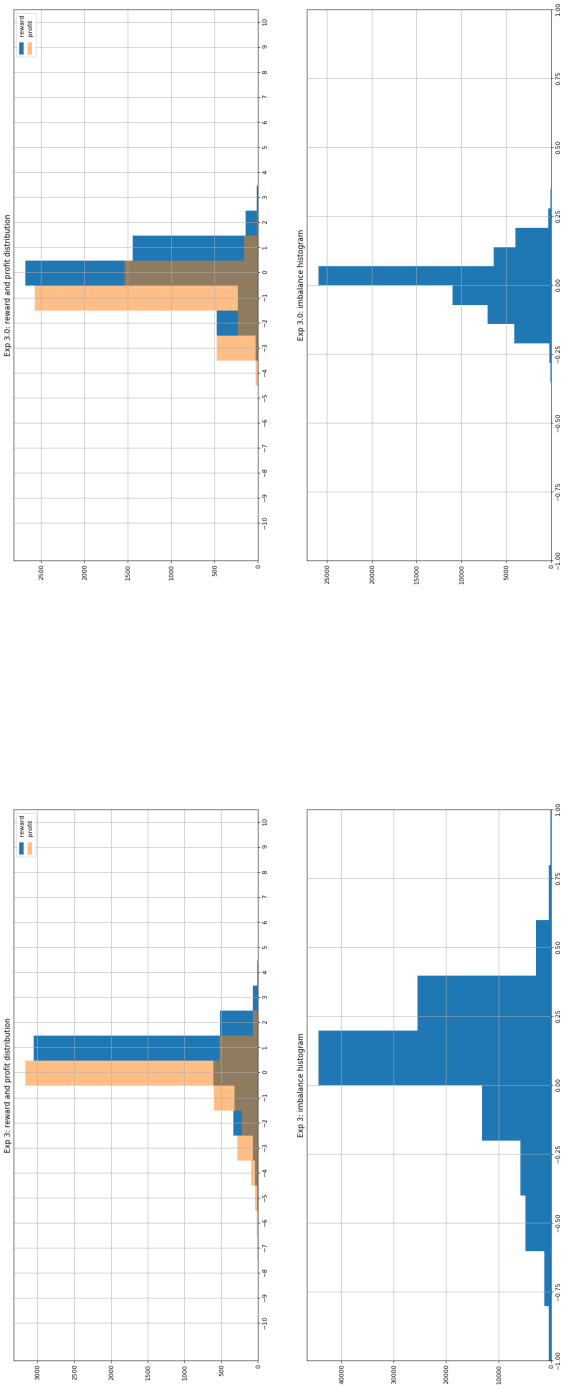
(a) (b)

Figure C.1: Reward and Imbalance distributions for experiments 1 and 10



(a) (b)

Figure C.2: Reward and Imbalance distributions for experiments 2 and 20



(a) (b)

Figure C.3: Reward and Imbalance distributions for experiments 3 and 30

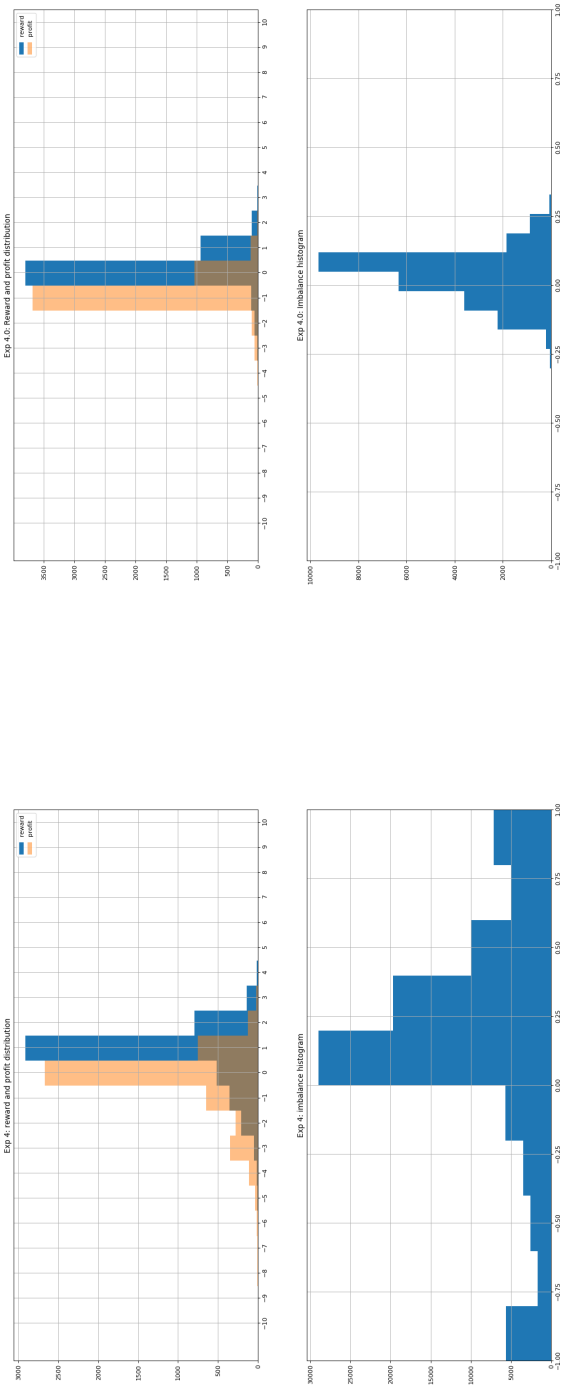


Figure C.4: Reward and Imbalance distributions for experiments 4 and 40

C.2 Chapter 4: Decision Trees

After training the trading policies for the various experimental configurations in Chapter 4, I trained a tree-classifier to interpret them. The input features were the input features of the Q-function (Action value function) and the predicted variable was the chosen action. The experiments appended ".0" correspond to those where the action space was restricted to only allow doing nothing and executing at best.

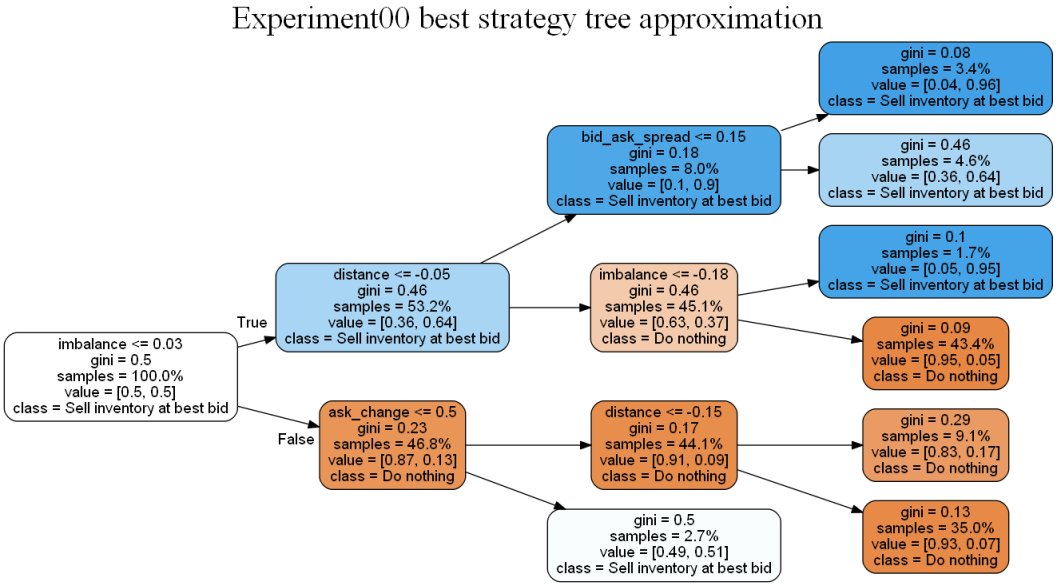


Figure C.5: Exp 0.0 tree classifier approximation: The strategy will sell once the best bid has improved by 2 or more from entry (profit taking), if best ask declines it also advocates selling (stop loss), otherwise it will wait.

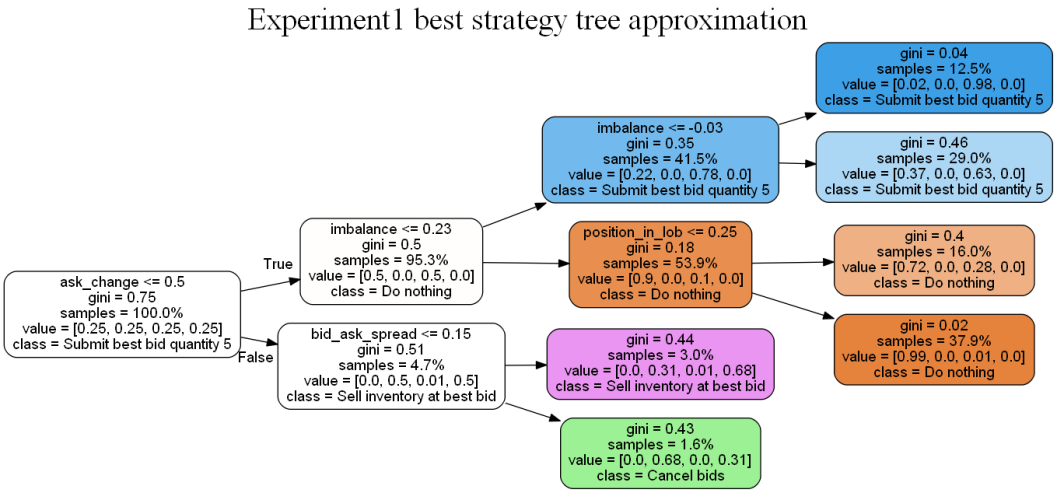


Figure C.6: Exp 1 tree classifier approximation: The upper part of the tree shows that adding orders to the bid occurs when imbalance is not high enough. The lower branch shows that order cancellation occurs when spread is small, which can indicate a higher risk of execution. Splitting first on ask change is in common with best strategy found in Exp0 (Figure 4.6)

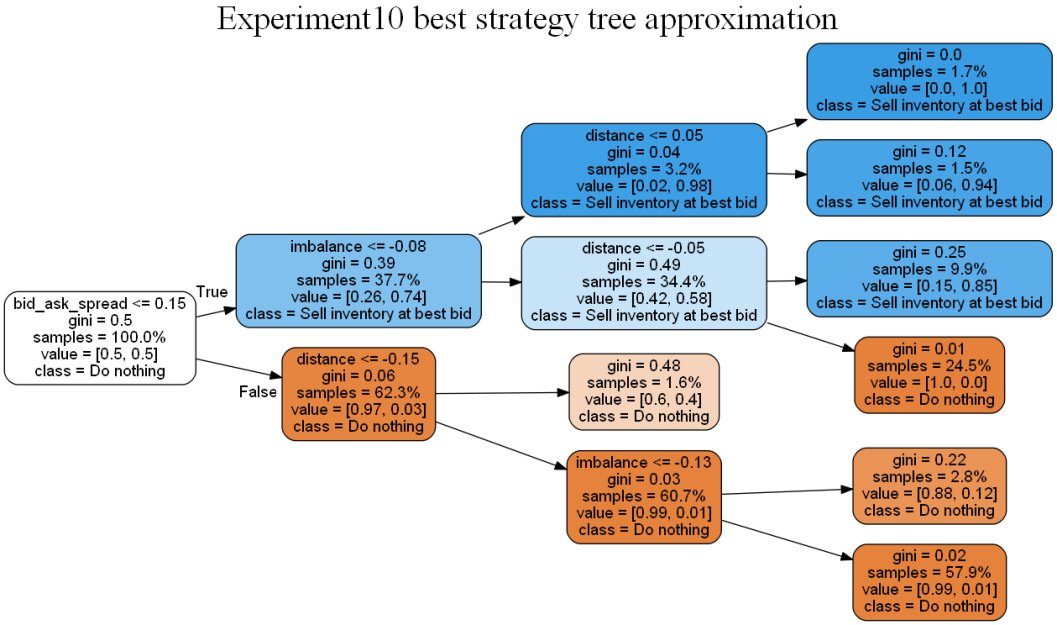


Figure C.7: Exp 1.0 tree classifier approximation: The strategy seems to learn that a negative imbalance is associated with negative markets and so exits position. Otherwise there is some profit taking element by the presence of distance as a splitting variable. A wide bid ask spread could be associated with stationary market movements hence decision to do nothing.

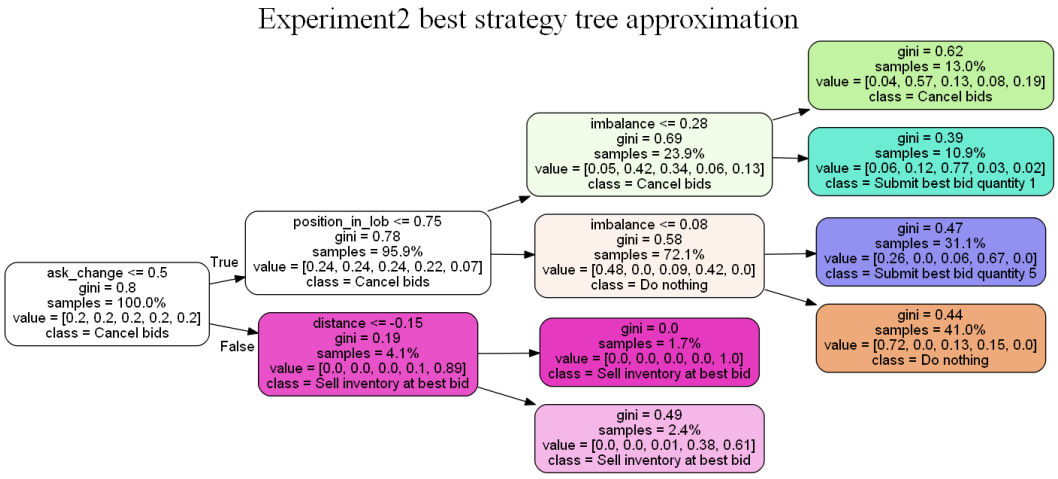


Figure C.8: Exp 2 tree classifier approximation: The strategy closes out the position after negative moves in best ask. Bids are cancelled if the agent gets too close to the front of the queue. Otherwise the strategy involves adding orders if the imbalance is not already above a threshold. This seems like a manipulative strategy.

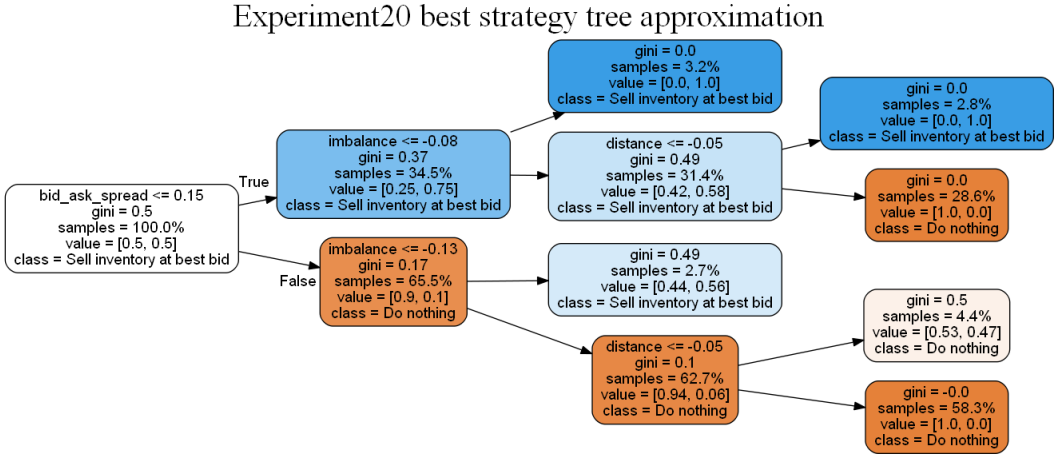


Figure C.9: Exp 2.0 tree classifier approximation: A rational profit taking decision can be seen in the upper branch. In the lower branch negative imbalance is a trigger to sell inventory.

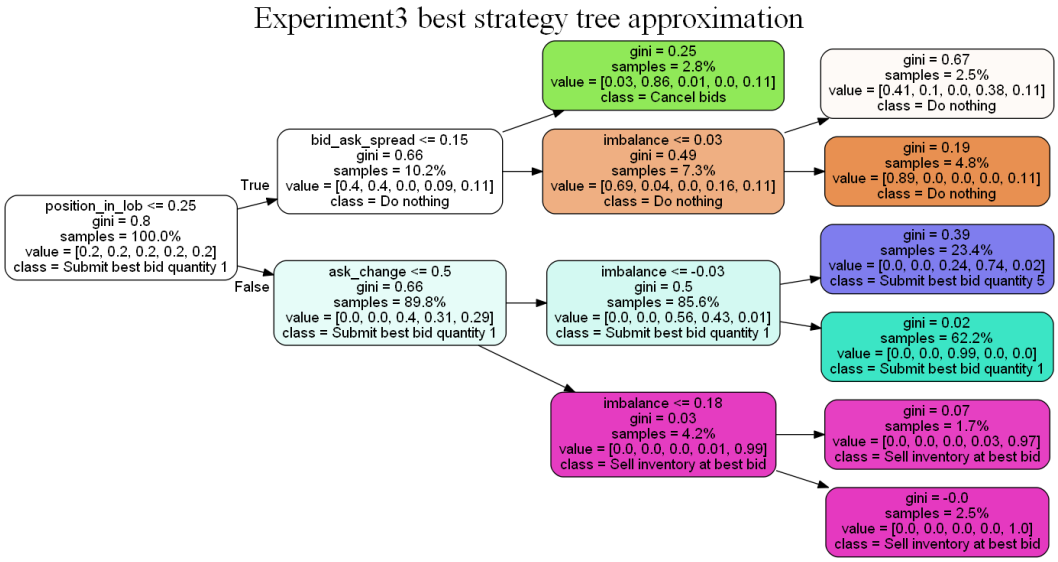


Figure C.10: Exp 3 tree classifier approximation: Cancelling bids when the bid ask spread and position in LOB are both small is rational. The lower half of the tree shows that the position is closed out if best ask is not declining. This might account for the strategy failing to get many high returns as shown in Figure 4.3

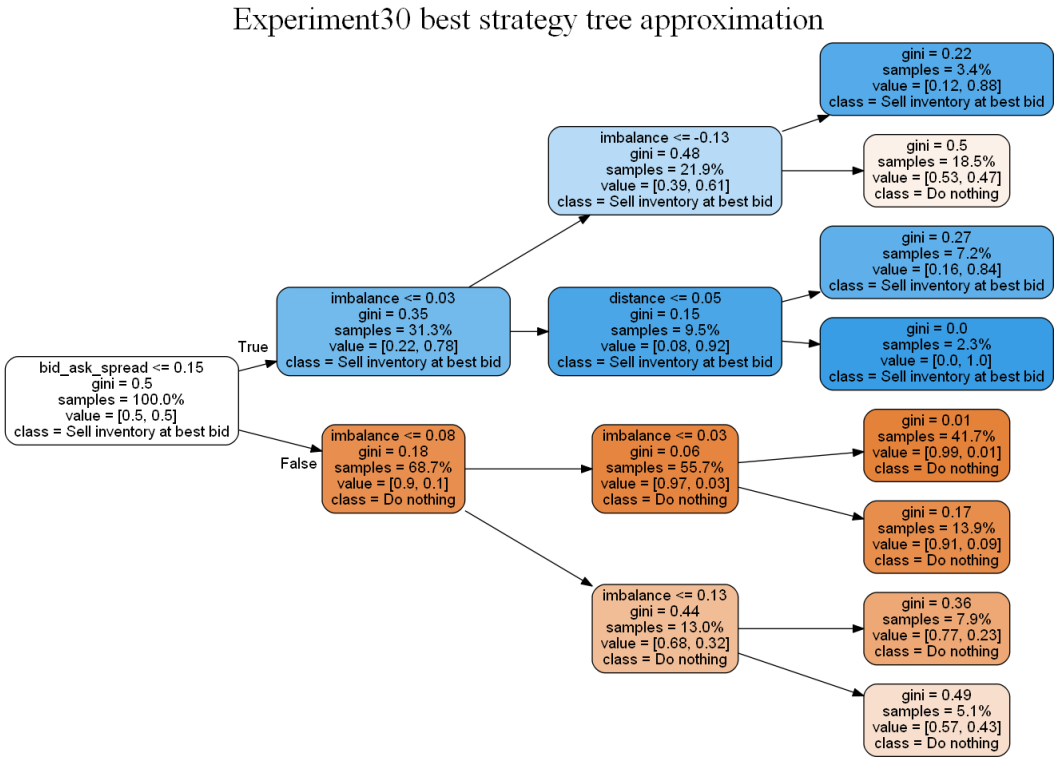


Figure C.11: Exp 3.0 tree classifier approximation: This tree is difficult to interpret.

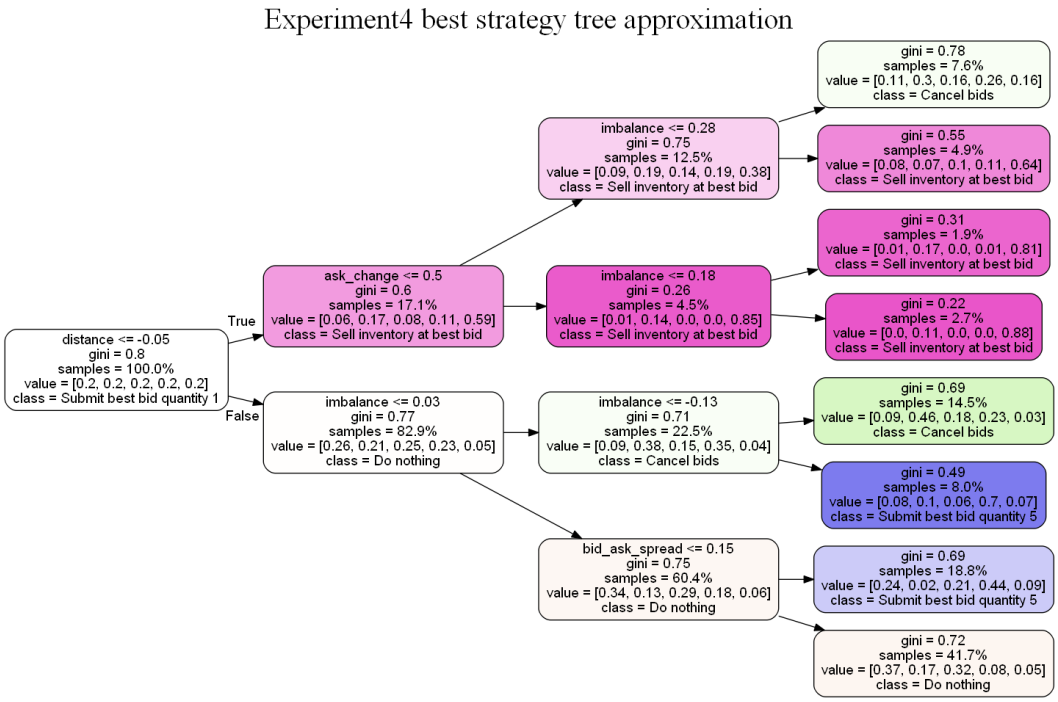


Figure C.12: Exp 4 tree classifier approximation: Here the classifier failed to 'explain' the strategy in the sense that the overall accuracy was only 0.49.

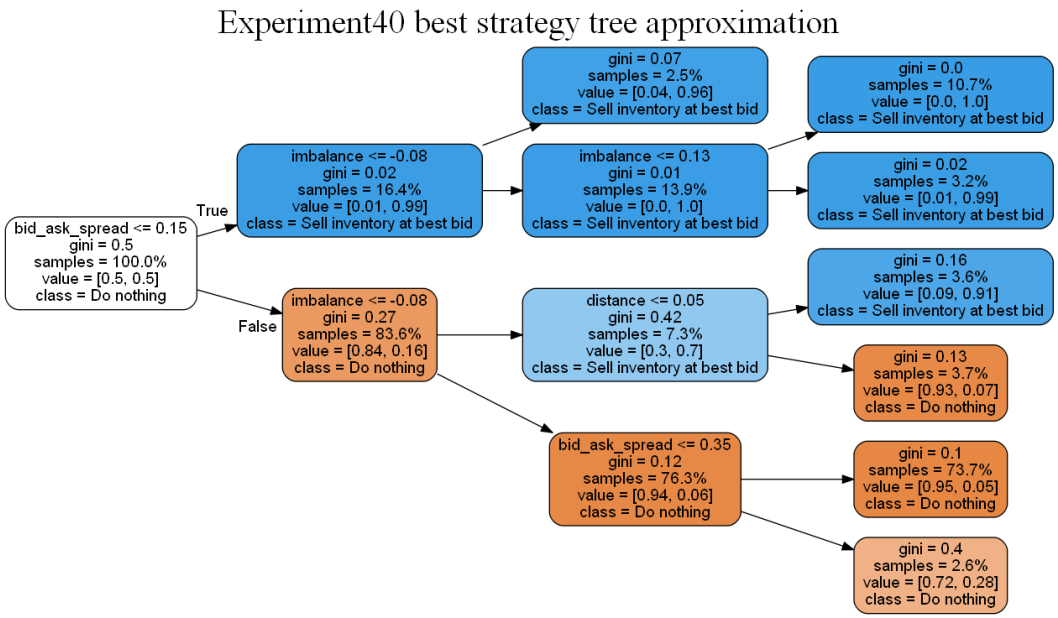


Figure C.13: Exp 4.0 tree classifier approximation: Whilst the classifier describes the strategy well (96% accuracy, Table 4.8), the learned strategy is not rational, deciding to always sell when bid ask spread is narrow.

D Chapter 6: Supplementary results

The section contains the full ANOVA and other related results for the experiments 4.1-3 in Chapter 6. They are included for completeness.

Cases	Sum of Squares	df	Mean Square	F	p	η^2	η_p^2	ω^2
Wind	6173.603	1	6173.603	270.929	< .001	0.458	0.688	0.583
Wind * AI	4.934	1	4.934	0.217	0.643	3.659e-4	0.002	0.000
Wind * Definition	2.312	1	2.312	0.101	0.751	1.714e-4	8.241e-4	0.000
Wind * AI * Definition	18.323	1	18.323	0.804	0.372	0.001	0.006	0.000
Residuals	2802.777	123	22.787					
Legal	292.129	1	292.129	63.497	< .001	0.022	0.340	0.118
Legal * AI	0.599	1	0.599	0.130	0.719	4.444e-5	0.001	0.000
Legal * Definition	5.805	1	5.805	1.262	0.263	4.305e-4	0.010	5.623e-4
Legal * AI * Definition	0.255	1	0.255	0.055	0.814	1.889e-5	4.498e-4	0.000
Residuals	565.885	123	4.601					
Benefit	140.232	1	140.232	34.362	< .001	0.010	0.218	0.062
Benefit * AI	0.042	1	0.042	0.010	0.919	3.106e-6	8.343e-5	0.000
Benefit * Definition	2.579	1	2.579	0.632	0.428	1.913e-4	0.005	0.000
Benefit * AI * Definition	1.331	1	1.331	0.326	0.569	9.872e-5	0.003	0.000
Residuals	501.968	123	4.081					
Wind * Legal	2.034	1	2.034	0.631	0.429	1.509e-4	0.005	0.000
Wind * Legal * AI	1.636	1	1.636	0.508	0.477	1.214e-4	0.004	0.000
Wind * Legal * Definition	0.405	1	0.405	0.126	0.723	3.007e-5	0.001	0.000
Wind * Legal * AI * Definition	3.367	1	3.367	1.045	0.309	2.497e-4	0.008	7.303e-5
Residuals	396.483	123	3.223					
Wind * Benefit	23.514	1	23.514	10.464	0.002	0.002	0.078	0.011
Wind * Benefit * AI	14.187	1	14.187	6.313	0.013	0.001	0.049	0.006
Wind * Benefit * Definition	1.432	1	1.432	0.637	0.426	1.062e-4	0.005	0.000
Wind * Benefit * AI * Definition	0.275	1	0.275	0.122	0.727	2.037e-5	9.929e-4	0.000
Residuals	276.410	123	2.247					
Legal * Benefit	7.181	1	7.181	2.415	0.123	5.326e-4	0.019	0.002
Legal * Benefit * AI	8.411	1	8.411	2.829	0.095	6.238e-4	0.022	0.003
Legal * Benefit * Definition	1.600	1	1.600	0.538	0.465	1.186e-4	0.004	0.000
Legal * Benefit * AI * Definition	5.983	1	5.983	2.012	0.159	4.437e-4	0.016	0.002
Residuals	365.712	123	2.973					
Wind * Legal * Benefit	19.894	1	19.894	8.913	0.003	0.001	0.068	0.009
Wind * Legal * Benefit * AI	0.325	1	0.325	0.146	0.703	2.410e-5	0.001	0.000
Wind * Legal * Benefit * Definition	4.730	1	4.730	2.119	0.148	3.508e-4	0.017	0.001
Wind * Legal * Benefit * AI * Definition	4.091	1	4.091	1.833	0.178	3.034e-4	0.015	0.001
Residuals	274.556	123	2.232					

Table D.1: Experiment 4.1 Within subject effects Anova. Significant effects highlighted.

Cases	Sum of Squares	df	Mean Square	F	p	η^2	η_p^2	ω^2
AI	0.562	1	0.562	0.044	0.834	4.167e-5	3.605e-4	0.000
Definition	0.050	1	0.050	0.004	0.950	3.718e-6	3.218e-5	0.000
AI * Definition	0.087	1	0.087	0.007	0.934	6.476e-6	5.605e-5	0.000
Residuals	1557.713	123	12.664					

Table D.2: Experiment 4.1: Between Subjects Effects

	F	df1	df2	p
Wind Legal Benefit	1.071	3	123	0.364
Wind Legal No-Benefit	0.429	3	123	0.732
Wind Not-Legal Benefit	1.261	3	123	0.291
Wind Not-Legal No-Benefit	4.335	3	123	0.006
No-Wind Legal Benefit	0.959	3	123	0.415
No-Wind Legal No-Benefit	1.505	3	123	0.217
No-Wind Not-Legal Benefit	3.334	3	123	0.022
No-Wind Not-Legal No-Benefit	0.408	3	123	0.747

Table D.3: Experiment 4.1 Levene's test for Equality of Variances within groups

Cases	Sum of Squares	df	Mean Square	F	p	η^2	η_p^2	ω^2
AI	45.264	1	45.264	7.441	0.007	0.001	0.055	0.010
AI * Definition	18.160	1	18.160	2.986	0.086	5.203e-4	0.023	0.003
Residuals	778.594	128	6.083					
Wind	7249.090	1	7249.090	143.559	< .001	0.208	0.529	0.430
Wind * Definition	1174.090	1	1174.090	23.251	< .001	0.034	0.154	0.105
Residuals	6463.449	128	50.496					
Legal	1944.762	1	1944.762	82.906	< .001	0.056	0.393	0.241
Legal * Definition	227.958	1	227.958	9.718	0.002	0.007	0.071	0.033
Residuals	3002.546	128	23.457					
Benefit	257.733	1	257.733	28.077	< .001	0.007	0.180	0.056
Benefit * Definition	1.179	1	1.179	0.128	0.721	3.377e-5	0.001	0.000
Residuals	1174.979	128	9.180					
AI * Wind	3.522	1	3.522	0.543	0.463	1.009e-4	0.004	0.000
AI * Wind * Definition	3.522	1	3.522	0.543	0.463	1.009e-4	0.004	0.000
Residuals	830.728	128	6.490					
AI * Legal	2.516e-4	1	2.516e-4	4.799e-5	0.994	7.208e-9	3.750e-7	0.000
AI * Legal * Definition	7.035	1	7.035	1.342	0.249	2.015e-4	0.010	4.826e-4
Residuals	671.023	128	5.242					
Wind * Legal	26.813	1	26.813	2.447	0.120	7.682e-4	0.019	0.004
Wind * Legal * Definition	38.667	1	38.667	3.529	0.063	0.001	0.027	0.006
Residuals	1402.538	128	10.957					
AI * Benefit	0.624	1	0.624	0.173	0.678	1.788e-5	0.001	0.000
AI * Benefit * Definition	4.417	1	4.417	1.226	0.270	1.265e-4	0.009	2.324e-4
Residuals	461.165	128	3.603					
Wind * Benefit	53.008	1	53.008	5.702	0.018	0.002	0.043	0.010
Wind * Benefit * Definition	3.958	1	3.958	0.426	0.515	1.134e-4	0.003	0.000
Residuals	1189.900	128	9.296					
Legal * Benefit	9.571	1	9.571	1.842	0.177	2.742e-4	0.014	0.001
Legal * Benefit * Definition	33.818	1	33.818	6.509	0.012	9.688e-4	0.048	0.008
Residuals	664.991	128	5.195					
AI * Wind * Legal	25.379	1	25.379	4.507	0.036	7.271e-4	0.034	0.005
AI * Wind * Legal * Definition	0.379	1	0.379	0.067	0.796	1.084e-5	5.249e-4	0.000
Residuals	720.722	128	5.631					
AI * Wind * Benefit	5.472	1	5.472	0.944	0.333	1.568e-4	0.007	0.000
AI * Wind * Benefit * Definition	1.522	1	1.522	0.263	0.609	4.359e-5	0.002	0.000
Residuals	741.740	128	5.795					
AI * Legal * Benefit	18.391	1	18.391	2.799	0.097	5.269e-4	0.021	0.003
AI * Legal * Benefit * Definition	3.637	1	3.637	0.554	0.458	1.042e-4	0.004	0.000
Residuals	841.017	128	6.570					
Wind * Legal * Benefit	2.178	1	2.178	0.255	0.614	6.240e-5	0.002	0.000
Wind * Legal * Benefit * Definition	9.559	1	9.559	1.120	0.292	2.739e-4	0.009	2.472e-4
Residuals	1092.595	128	8.536					
AI * Wind * Legal * Benefit	8.591	1	8.591	1.663	0.200	2.461e-4	0.013	9.243e-4
AI * Wind * Legal * Benefit * Definition	0.803	1	0.803	0.155	0.694	2.300e-5	0.001	0.000
Residuals	661.248	128	5.166					

Table D.4: Experiment 4.2 Repeated measures ANOVA: Within Subjects Effects. Significant effects highlighted.

Cases	Sum of Squares	df	Mean Square	F	p	η^2	η_p^2	ω^2
Definition	16.361	1	16.361	0.695	0.406	4.687e-4	0.005	0.000
Residuals	3012.478	128	23.535					

Table D.5: Experiment 4.2 Repeated measures ANOVA: between Subjects Effects

	F	df1	df2	p
AI Wind Legal Benefit	0.925	1	128	0.338
AI Wind Legal No-Benefit	3.924	1	128	0.050
AI Wind Not-legal Benefit	6.009	1	128	0.016
AI Wind Not-legal Not Benefit	6.967	1	128	0.009
AI No-Wind Legal Benefit	10.954	1	128	0.001
AI No-Wind Legal NotBenefit	11.119	1	128	0.001
AI No-Wind Not Illegal Benefit	58.103	1	128	< .001
AI No-Wind Not Illegal NoBenefit	6.130	1	128	0.015
Hum Wind Legal Benefit	0.026	1	128	0.873
Hum Wind Legal No-Benefit	2.223	1	128	0.138
Hum Wind Not-legal Benefit	0.724	1	128	0.396
Hum Wind Not-legal No-Benefit	7.763	1	128	0.006
Hum No-Wind Legal Benefit	10.904	1	128	0.001
Hum No-Wind Legal No-Benefit	3.874	1	128	0.051
Hum No-Wind Not-legal Benefit	34.390	1	128	< .001
Hum No-Wind Not-legal No-Benefit	14.782	1	128	< .001

Table D.6: Experiment 4.2 Levene's test for Equality of Variances

		95% CI for Mean Difference						
Variable	Comparison	Estimate	Lower	Upper	SE	df	t	p
Wind	False - True	3.782	3.119	4.352	0.337	129	11.208	< .001
Legal	True - False	1.914	1.514	2.355	0.219	129	8.724	< .001
Benefit	True - False	0.703	0.441	0.967	0.132	129	5.308	< .001
Pilot	AI - Human	-0.289	-0.509	-0.081	0.109	129	-2.656	0.009

Table D.7: Experiment 4.2 Contrasts. Intent scores are averaged across groups not being contrasted. The t-test variant does not assume equal variances.

Cases	Sum of Squares	df	Mean Square	F	p	η^2	η_p^2	ω^2
Defn	6.826	1	6.826	0.708	0.403	3.672e-4	0.010	0.000
Defn * pilot	21.144	1	21.144	2.192	0.143	0.001	0.030	0.005
Defn * F_D	5.132	1	5.132	0.532	0.468	2.761e-4	0.008	0.000
Defn * pilot * F_D	10.098	1	10.098	1.047	0.310	5.433e-4	0.015	2.103e-4
Residuals	675.207	70	9.646					
Wind	4229.268	1	4229.268	98.457	< .001	0.228	0.584	0.481
Wind * pilot	0.518	1	0.518	0.012	0.913	2.789e-5	1.724e-4	0.000
Wind * F_D	109.377	1	109.377	2.546	0.115	0.006	0.035	0.015
Wind * pilot * F_D	37.495	1	37.495	0.873	0.353	0.002	0.012	0.000
Residuals	3006.874	70	42.955					
Legal	513.019	1	513.019	28.377	< .001	0.028	0.288	0.153
Legal * pilot	1.713	1	1.713	0.095	0.759	9.215e-5	0.001	0.000
Legal * F_D	53.105	1	53.105	2.937	0.091	0.003	0.040	0.013
Legal * pilot * F_D	1.166	1	1.166	0.065	0.800	6.275e-5	9.207e-4	0.000
Residuals	1265.505	70	18.079					
Benefit	108.760	1	108.760	11.627	0.001	0.006	0.142	0.045
Benefit * pilot	5.082	1	5.082	0.543	0.464	2.734e-4	0.008	0.000
Benefit * F_D	0.743	1	0.743	0.079	0.779	3.999e-5	0.001	0.000
Benefit * pilot * F_D	15.886	1	15.886	1.698	0.197	8.547e-4	0.024	0.003
Residuals	654.789	70	9.354					
Defn * Wind	14.032	1	14.032	0.855	0.358	7.550e-4	0.012	0.000
Defn * Wind * pilot	0.558	1	0.558	0.034	0.854	2.999e-5	4.848e-4	0.000
Defn * Wind * F_D	470.956	1	470.956	28.684	< .001	0.025	0.291	0.147
Defn * Wind * pilot * F_D	58.476	1	58.476	3.562	0.063	0.003	0.048	0.016
Residuals	1149.324	70	16.419					
Defn * Legal	112.422	1	112.422	10.898	0.002	0.006	0.135	0.044
Defn * Legal * pilot	0.609	1	0.609	0.059	0.809	3.275e-5	8.423e-4	0.000
Defn * Legal * F_D	0.896	1	0.896	0.087	0.769	4.821e-5	0.001	0.000
Defn * Legal * pilot * F_D	6.566	1	6.566	0.637	0.428	3.533e-4	0.009	0.000
Residuals	722.125	70	10.316					
Wind * Legal	67.322	1	67.322	10.766	0.002	0.004	0.133	0.031
Wind * Legal * pilot	33.386	1	33.386	5.339	0.024	0.002	0.071	0.014
Wind * Legal * F_D	1.854	1	1.854	0.297	0.588	9.976e-5	0.004	0.000
Wind * Legal * pilot * F_D	0.061	1	0.061	0.010	0.922	3.273e-6	1.389e-4	0.000
Residuals	437.723	70	6.253					
Defn * Benefit	3.550	1	3.550	0.521	0.473	1.910e-4	0.007	0.000
Defn * Benefit * pilot	1.553	1	1.553	0.228	0.635	8.354e-5	0.003	0.000
Defn * Benefit * F_D	11.107	1	11.107	1.630	0.206	5.976e-4	0.023	0.002
Defn * Benefit * pilot * F_D	6.633	1	6.633	0.973	0.327	3.569e-4	0.014	0.000
Residuals	477.136	70	6.816					
Wind * Benefit	9.609	1	9.609	1.337	0.251	5.170e-4	0.019	0.001
Wind * Benefit * pilot	0.685	1	0.685	0.095	0.758	3.688e-5	0.001	0.000
Wind * Benefit * F_D	18.035	1	18.035	2.510	0.118	9.703e-4	0.035	0.005
Wind * Benefit * pilot * F_D	20.064	1	20.064	2.792	0.099	0.001	0.038	0.006
Residuals	503.019	70	7.186					
Legal * Benefit	0.158	1	0.158	0.031	0.861	8.474e-6	4.440e-4	0.000
Legal * Benefit * pilot	0.061	1	0.061	0.012	0.913	3.303e-6	1.731e-4	0.000
Legal * Benefit * F_D	5.962	1	5.962	1.177	0.282	3.208e-4	0.017	4.919e-4
Legal * Benefit * pilot * F_D	10.972	1	10.972	2.166	0.146	5.903e-4	0.030	0.003
Residuals	354.584	70	5.065					
Defn * Wind * Legal	9.737	1	9.737	1.892	0.173	5.239e-4	0.026	0.003
Defn * Wind * Legal * pilot	0.776	1	0.776	0.151	0.699	4.173e-5	0.002	0.000
Defn * Wind * Legal * F_D	11.753	1	11.753	2.283	0.135	6.323e-4	0.032	0.004
Defn * Wind * Legal * pilot * F_D	0.155	1	0.155	0.030	0.863	8.316e-6	4.288e-4	0.000
Residuals	360.295	70	5.147					
Defn * Wind * Benefit	0.040	1	0.040	0.018	0.894	2.129e-6	2.540e-4	0.000
Defn * Wind * Benefit * pilot	7.212	1	7.212	3.241	0.076	3.880e-4	0.044	0.003
Defn * Wind * Benefit * F_D	3.827	1	3.827	1.720	0.194	2.059e-4	0.024	9.874e-4
Defn * Wind * Benefit * pilot * F_D	3.757	1	3.757	1.689	0.198	2.021e-4	0.024	9.445e-4
Residuals	155.738	70	2.225					
Defn * Legal * Benefit	0.496	1	0.496	0.090	0.765	2.667e-5	0.001	0.000
Defn * Legal * Benefit * pilot	10.179	1	10.179	1.852	0.178	5.477e-4	0.026	0.003
Defn * Legal * Benefit * F_D	2.106	1	2.106	0.383	0.538	1.133e-4	0.005	0.000
Defn * Legal * Benefit * pilot * F_D	8.180	1	8.180	1.488	0.227	4.401e-4	0.021	0.001
Residuals	384.738	70	5.496					
Wind * Legal * Benefit	46.703	1	46.703	6.949	0.010	0.003	0.090	0.020
Wind * Legal * Benefit * pilot	0.537	1	0.537	0.080	0.778	2.890e-5	0.001	0.000
Wind * Legal * Benefit * F_D	0.020	1	0.020	0.003	0.956	1.095e-6	4.324e-5	0.000
Wind * Legal * Benefit * pilot * F_D	3.804	1	3.804	0.566	0.454	2.047e-4	0.008	0.000
Residuals	470.472	70	6.721					
Defn * Wind * Legal * Benefit	3.716	1	3.716	0.953	0.332	1.999e-4	0.013	0.000
Defn * Wind * Legal * Benefit * pilot	16.817	1	16.817	4.312	0.042	9.048e-4	0.058	0.007
Defn * Wind * Legal * Benefit * F_D	5.253	1	5.253	1.347	0.250	2.826e-4	0.019	7.775e-4
Defn * Wind * Legal * Benefit * pilot * F_D	0.389	1	0.389	0.100	0.753	2.094e-5	0.001	0.000
Residuals	272.979	70	3.900					

Table D.8: Experiment 4.3: Intent within Subjects Effects. F_D group refers to whether participants saw formal definition for first set of 8 questions, or were asked to use their own definition of intent.

Cases	Sum of Squares	df	Mean Square	F	p	η^2	η_p^2	ω^2
pilot	125.162	1	125.162	6.075	0.016	0.007	0.080	0.035
F_D	8.017	1	8.017	0.389	0.535	4.313e-4	0.006	0.000
pilot * F_D	10.517	1	10.517	0.510	0.477	5.658e-4	0.007	0.000
Residuals	1442.144	70	20.602					

Table D.9: Experiment 4.3: Intent between Subjects Effects

Cases	Sum of Squares	df	Mean Square	F	p	η^2	η_p^2	ω^2
Error type	39.278	1	39.278	7.471	0.008	0.053	0.094	0.046
Error type * Pilot ID	4.900	1	4.900	0.932	0.338	0.007	0.013	0.000
Residuals	378.539	72	5.257					

Table D.10: Experiment 4.3: Error Attribution Repeated Measures ANOVA: Within Subjects Effects

Cases	Sum of Squares	df	Mean Square	F	p	η^2	η_p^2	ω^2
Pilot ID	0.133	1	0.133	0.030	0.862	1.797e-4	4.197e-4	0.000
Residuals	316.387	72	4.394					

Table D.11: Experiment 4.3: Error Attribution Repeated Measures ANOVA: Between Subjects Effects

		95% CI for Mean Difference					
Comparison	Estimate	Lower	Upper	SE	df	t	p
Drone err - Pilot err	-1.031	-1.782	-0.279	0.377	72	-2.733	0.008
Human grp - AI grp	0.060	-0.627	0.747	0.345	72	0.174	0.862

Table D.12: Experiment 4.3 Error Attribution

Cases	Sum of Squares	df	Mean Square	F	p	η^2	η_p^2	ω^2
Defn	7.241	1	7.241	1.367	0.246	6.107e-4	0.019	7.485e-4
Defn * F_D	11.437	1	11.437	2.158	0.146	9.646e-4	0.030	0.002
Defn * pilot	1.301	1	1.301	0.245	0.622	1.097e-4	0.003	0.000
Defn * F_D * pilot	6.662	1	6.662	1.257	0.266	5.618e-4	0.018	5.254e-4
Residuals	370.912	70	5.299					
Wind	3171.921	1	3171.921	111.312	< .001	0.268	0.614	0.426
Wind * F_D	1.179	1	1.179	0.041	0.839	9.945e-5	5.908e-4	0.000
Wind * pilot	0.281	1	0.281	0.010	0.921	2.373e-5	1.410e-4	0.000
Wind * F_D * pilot	80.868	1	80.868	2.838	0.097	0.007	0.039	0.012
Residuals	1994.708	70	28.496					
Legal	444.139	1	444.139	71.620	< .001	0.037	0.506	0.142
Legal * F_D	2.390	1	2.390	0.385	0.537	2.015e-4	0.005	0.000
Legal * pilot	15.036	1	15.036	2.425	0.124	0.001	0.033	0.003
Legal * F_D * pilot	3.729e-4	1	3.729e-4	6.013e-5	0.994	3.145e-8	8.590e-7	0.000
Residuals	434.093	70	6.201					
Benefit	35.840	1	35.840	9.306	0.003	0.003	0.117	0.013
Benefit * F_D	1.966	1	1.966	0.511	0.477	1.658e-4	0.007	0.000
Benefit * pilot	1.178	1	1.178	0.306	0.582	9.934e-5	0.004	0.000
Benefit * F_D * pilot	0.293	1	0.293	0.076	0.783	2.474e-5	0.001	0.000
Residuals	269.584	70	3.851					
Defn * Wind	10.852	1	10.852	2.224	0.140	9.152e-4	0.031	0.002
Defn * Wind * F_D	14.837	1	14.837	3.040	0.086	0.001	0.042	0.004
Defn * Wind * pilot	1.841	1	1.841	0.377	0.541	1.553e-4	0.005	0.000
Defn * Wind * F_D * pilot	1.339	1	1.339	0.274	0.602	1.129e-4	0.004	0.000
Residuals	341.619	70	4.880					
Defn * Legal	9.091	1	9.091	3.090	0.083	7.667e-4	0.042	0.003
Defn * Legal * F_D	0.272	1	0.272	0.092	0.762	2.291e-5	0.001	0.000
Defn * Legal * pilot	1.371	1	1.371	0.466	0.497	1.156e-4	0.007	0.000
Defn * Legal * F_D * pilot	0.198	1	0.198	0.067	0.796	1.668e-5	9.595e-4	0.000
Residuals	205.973	70	2.942					
Wind * Legal	39.218	1	39.218	8.393	0.005	0.003	0.107	0.013
Wind * Legal * F_D	0.249	1	0.249	0.053	0.818	2.104e-5	7.621e-4	0.000
Wind * Legal * pilot	0.760	1	0.760	0.163	0.688	6.413e-5	0.002	0.000
Wind * Legal * F_D * pilot	9.606	1	9.606	2.056	0.156	8.101e-4	0.029	0.002
Residuals	327.106	70	4.673					
Defn * Benefit	0.809	1	0.809	0.253	0.616	6.822e-5	0.004	0.000
Defn * Benefit * F_D	0.086	1	0.086	0.027	0.870	7.219e-6	3.824e-4	0.000
Defn * Benefit * pilot	0.234	1	0.234	0.073	0.787	1.976e-5	0.001	0.000
Defn * Benefit * F_D * pilot	5.773	1	5.773	1.806	0.183	4.869e-4	0.025	0.001
Residuals	223.732	70	3.196					
Wind * Benefit	41.050	1	41.050	14.272	< .001	0.003	0.169	0.016
Wind * Benefit * F_D	0.003	1	0.003	0.001	0.973	2.785e-7	1.640e-5	0.000
Wind * Benefit * pilot	8.764	1	8.764	3.047	0.085	7.392e-4	0.042	0.002
Wind * Benefit * F_D * pilot	4.415	1	4.415	1.535	0.220	3.723e-4	0.021	6.351e-4
Residuals	201.331	70	2.876					
Legal * Benefit	6.472	1	6.472	1.998	0.162	5.458e-4	0.028	0.001
Legal * Benefit * F_D	3.835	1	3.835	1.184	0.280	3.234e-4	0.017	2.433e-4
Legal * Benefit * pilot	2.093	1	2.093	0.646	0.424	1.766e-4	0.009	0.000
Legal * Benefit * F_D * pilot	0.070	1	0.070	0.022	0.883	5.914e-6	3.091e-4	0.000
Residuals	226.754	70	3.239					
Defn * Wind * Legal	0.887	1	0.887	0.337	0.563	7.484e-5	0.005	0.000
Defn * Wind * Legal * F_D	2.422e-4	1	2.422e-4	9.208e-5	0.992	2.043e-8	1.315e-6	0.000
Defn * Wind * Legal * pilot	2.788	1	2.788	1.060	0.307	2.351e-4	0.015	6.563e-5
Defn * Wind * Legal * F_D * pilot	1.711	1	1.711	0.651	0.423	1.443e-4	0.009	0.000
Residuals	184.120	70	2.630					
Defn * Wind * Benefit	0.548	1	0.548	0.174	0.677	4.621e-5	0.002	0.000
Defn * Wind * Benefit * F_D	1.276	1	1.276	0.406	0.526	1.076e-4	0.006	0.000
Defn * Wind * Benefit * pilot	0.164	1	0.164	0.052	0.820	1.381e-5	7.443e-4	0.000
Defn * Wind * Benefit * F_D * pilot	6.196	1	6.196	1.973	0.165	5.225e-4	0.027	0.001
Residuals	219.816	70	3.140					
Defn * Legal * Benefit	0.892	1	0.892	0.259	0.612	7.523e-5	0.004	0.000
Defn * Legal * Benefit * F_D	0.583	1	0.583	0.169	0.682	4.919e-5	0.002	0.000
Defn * Legal * Benefit * pilot	0.861	1	0.861	0.250	0.619	7.261e-5	0.004	0.000
Defn * Legal * Benefit * F_D * pilot	2.365	1	2.365	0.687	0.410	1.995e-4	0.010	0.000
Residuals	241.028	70	3.443					
Wind * Legal * Benefit	17.803	1	17.803	6.081	0.016	0.002	0.080	0.006
Wind * Legal * Benefit * F_D	0.013	1	0.013	0.004	0.948	1.067e-6	6.175e-5	0.000
Wind * Legal * Benefit * pilot	0.661	1	0.661	0.226	0.636	5.579e-5	0.003	0.000
Wind * Legal * Benefit * F_D * pilot	2.922e-4	1	2.922e-4	9.981e-5	0.992	2.465e-8	1.426e-6	0.000
Residuals	204.944	70	2.928					
Defn * Wind * Legal * Benefit	5.926	1	5.926	1.974	0.164	4.998e-4	0.027	0.001
Defn * Wind * Legal * Benefit * F_D	0.634	1	0.634	0.211	0.647	5.345e-5	0.003	0.000
Defn * Wind * Legal * Benefit * pilot	2.823	1	2.823	0.940	0.336	2.380e-4	0.013	0.000
Defn * Wind * Legal * Benefit * F_D * pilot	0.951	1	0.951	0.317	0.575	8.018e-5	0.005	0.000
Residuals	210.107	70	3.002					

Table D.13: Experiment 4.3: Causal ratings, within Subjects Effects

Cases	Sum of Squares	df	Mean Square	F	p	η^2	η_p^2	ω^2
F_D	7.196	1	7.196	0.231	0.633	6.069e-4	0.003	0.000
pilot	16.487	1	16.487	0.528	0.470	0.001	0.007	0.000
F_D * pilot	0.275	1	0.275	0.009	0.926	2.316e-5	1.257e-4	0.000
Residuals	2185.289	70	31.218					

Table D.14: Experiment 4.3: Causal ratings, between Subjects Effects

Cases	Sum of Squares	df	Mean Square	F	p	η^2	η_p^2	ω^2
employee	150.979	1	150.979	14.897	< .001	0.016	0.016	0.015
role	8.332	1	8.332	0.822	0.365	8.821e-4	9.027e-4	0.000
Defn	9.509	2	4.754	0.469	0.626	0.001	0.001	0.000
employee * role	5.141	1	5.141	0.507	0.477	5.442e-4	5.571e-4	0.000
employee * Defn	32.538	2	16.269	1.605	0.201	0.003	0.004	0.001
role * Defn	10.274	2	5.137	0.507	0.603	0.001	0.001	0.000
employee * role * Defn	6.693	2	3.346	0.330	0.719	7.085e-4	7.251e-4	0.000
Residuals	9222.558	910	10.135					

Table D.15: ANOVA - Responsibility for harm caused by a pilot, ratings taken over Experiments 4.1,2 and 3

Experiment	Pilot	Definition	No	Unsure	Yes	**Total**	%No	%Unsure	%Yes
1	AI	The Formal	9	8	14	**31**	29	26	45
1	AI	Your	9	8	17	**34**	26	24	50
2	Both	The Formal	26	14	23	**63**	41	22	37
2	Both	your	21	9	37	**67**	31	13	55
3	AI	Both	13	3	20	**36**	36	8	56
3	Human	Both	11	4	23	**38**	29	11	61
Total			**89**	**46**	**134**	269			

Table D.16: Experiments 4.1,2,3: Response to question: Do you think AI can have intent?

Bibliography

Abbas, B., Belatreche, A., and Ahmed, B. (2018). Stock Price Manipulation Detection Using Empirical Mode Decomposition Based Kernel Density Estimation Clustering Method. *Advances in intelligent systems and computing*, 869:851–866.

Abbott, R. (2020). Reasonable Robots. *The Reasonable Robot*, pages 50–70.

Abbott, R. and Sarch, A. (2020). Punishing Artificial Intelligence: Legal Fiction or Science Fiction. *Is Law Computable?*, pages 323–384.

ACCC (2005). Predatory Pricing. Technical Report 5, Australian Competition and Consumer Commission.

Achiam, J., Held, D., Tamar, A., and Abbeel, P. (2017). Constrained Policy Optimization. *ICML*.

Adams, F. and Steadman, A. (2004). Intentional Action in Ordinary Language : Core Concept or Pragmatic Understanding? *Analysis*, 64(2):173–181.

Alexander, L. and Kessler, K. D. (1997). Mens Rea and Inchoate Crimes. *Journal of Criminal Law and Criminology*, 87(4):1138.

Alfonsi, A. and Acevedo, J. I. (2014a). Optimal Execution and Price Manipulations in Time-varying Limit Order Books. *Applied Mathematical Finance*, 21(3):201–237.

Alfonsi, A. and Acevedo, J. I. (2014b). Optimal Execution and Price Manipulations in Time-varying Limit Order Books. *Applied Mathematical Finance*, 21(3):201–237.

Alicke, M. D., Rose, D., and Bloom, D. (2011). Causation, Norm Violation and Culpable Control. *Journal of Philosophy*, 108(12):670–696.

Alldredge, J. (2015). The " CSI Effect " and Its Potential Impact on Juror Decisions. *Themis: Research Journal of Justice Studies and Forensic Science*, 3.

Alldridge, P. (1990). The Doctrine of Innocent Agency. *Criminal Law Forum*, 2(1):45–83.

Allen, F. and Gale, D. (1992). Stock-Price Manipulation. *The Review of Financial Studies*, 5(3):503–529.

Alshiekh, M., Bloem, R., Ehlers, R., Koenighofer, B., Niekum, S., and Topcu, U. (2017). Safe Reinforcement Learning via Shielding. arXivID: 1708.08611.

Alves, G. V., Dennis, L., and Fisher, M. (2020a). Formalisation and Implementation of Road Junction Rules on an Autonomous Vehicle Modelled as an Agent. In Sekerinski, E., Moreira, N., Oliveira, J. N., Ratiu, D., Guidotti, R., Farrell, M., Luckcuck, M., Marmsoler, D., Campos, J., Astarte, T., Gonnord, L., Cerone, A., Couto, L., Dongol, B., Kutrib, M., Monteiro, P., and Delmas, D., editors, *Formal Methods. FM 2019 International Workshops*, volume 12232, pages 217–232. Springer International Publishing, Cham. Series Title: Lecture Notes in Computer Science.

Alves, T. W., Florescu, I., Calhoun, G., and Bozdog, D. (2020b). SHIFT: A Highly Realistic Financial Market Simulation Platform. arXivID: 2002.11158.

Amodei, D., Olah, C., Steinhardt, J., Christiano, P., Schulman, J., and ManÃĺ,

D. (2016). Concrete Problems in AI Safety. *arXiv:1606.06565 [cs]*. arXiv: 1606.06565.

Amrouni, S., Moulin, A., Vann, J., Vyetrenko, S., Balch, T., and Veloso, M. (2021). ABIDES-gym: gym environments for multi-agent discrete event simulation and application to financial markets. In *Proceedings of the Second ACM International Conference on AI in Finance*, pages 1–9, Virtual Event. ACM.

An, J. and Cho, S. (2015). Variational Autoencoder based Anomaly Detection using Reconstruction Probability. In *Special Lecture on IE*.

Anjomshoae, S., Najjar, A., Calvaresi, D., and FrÃďmling, K. (2019). Explainable Agents and Robots: Results from a Systematic Literature Review. In *Proceedings of the 18th International Conference on Autonomous Agents andMultiAgent Systems*, Montreal.

Ashton, H. (2020). AI Legal Counsel to train and regulate legally constrained Autonomous systems. *IEEE International Conference on Big Data (Big Data)*, pages 2093–2098.

Ashton, H. (2021a). Causal Campbell-Goodhart's Law and Reinforcement Learning:. In *Proceedings of the 13th International Conference on Agents and Artificial Intelligence (ICAART)*, pages 67–73. SCITEPRESS - Science and Technology Publications.

Ashton, H. (2021b). Extending counterfactual accounts of intent to include oblique intent. axXivID: 2106.03684.

Ashton, H. (2021c). What criminal and civil law tells us about Safe RL techniques to generate law-abiding behaviour. *Workshop on AI Safety 2021 co-located with the Thirty Fifth AAAI Conference on Artificial Intelligence*.

Ashton, H. (2022a). Defining and Identifying the Legal Culpability of Side

Effects Using Causal Graphs. In *Workshop on AI Safety 2022 (SafeAI 2022) co-located with the Thirty Sixth AAAI Conference on Artificial Intelligence*.

Ashton, H. (2022b). Definitions of intent suitable for algorithms. *Artificial Intelligence and Law*.

Ashton, H. and Franklin, M. (2022a). A method to check that participants are imagining artificial minds when ascribing mental states. In *Proceedings of 24th Human Computer Interaction International conference (HCII)*.

Ashton, H. and Franklin, M. (2022b). The problem of behaviour and preference manipulation in AI systems. In *Workshop on AI Safety 2022 (SafeAI 2022) co-located with the Thirty Sixth AAAI Conference on Artificial Intelligence*.

Auer, P. and Ortner, R. (2010). UCB revisited: Improved regret bounds for the stochastic multi-armed bandit problem. *Periodica Mathematica Hungarica*, 61(1):55–65.

Bai, Y., Lam, H., Vyetrenko, S., and Balch, T. (2021). Efficient Calibration of Multi-Agent Market Simulators from Time Series with Bayesian Optimization. arXivID :2112.03874.

Baier, C. and Katoen, J.-P. (2008). *Principles Of Model Checking*. MIT Press.

Banks, J. (2019). A perceived moral agency scale: Development and validation of a metric for humans and social machines. *Computers in Human Behavior*, 90(November 2017):363–371.

Barber, D. (2016). Bayesian Reasoning and Machine Learning. *Machine Learning*, page 646.

Barreno, M., Nelson, B., Sears, R., Joseph, A. D., and Tygar, J. D. (2006). Can machine learning be secure? *ACM Symposium on Information, computer and communications security*, pages 16–25.

Bathaee, Y. (2018). The Artificial Intelligence Black Box and the Failure of Intent and Causation. *Harvard Journal of Law and Technology*, 31(2):890–938.

Belcak, P., Calliess, J.-P., and Zohren, S. (2020). Fast Agent-Based Simulation Framework of Limit Order Books with Applications to Pro-Rata Markets and the Study of Latency Effects. arXivID: 2008.07871.

Bench-Capon, T. (2020). Ethical approaches and autonomous systems. *Artificial Intelligence*, 281:103239.

Bentham, J. (1823). An Introduction to the Principles of Morals and Legislation. Prepared by Jonathan Bennett.

Bigman, Y. E. and Gray, K. (2018). People are averse to machines making moral decisions. *Cognition*, 181(March):21–34.

Boeckle, M., Schiestl, M., Frohnwieser, A., Gruber, R., Miller, R., Suddendorf, T., Gray, R. D., Taylor, A. H., and Clayton, N. S. (2020). New Caledonian crows plan for specific future tool use. *Proceedings of the Royal Society B: Biological Sciences*, 287(1938).

Borrageiro, G., Firoozye, N., and Barucca, P. (2022). Reinforcement Learning for Systematic FX Trading. *IEEE Access*, 10:5024–5036.

Bratman, M. E. (1990). What is Intention? In Cohen, P. R., Morgan, J., and Pollock, M. E., editors, *Intentions in communication*, chapter 2. MIT Press.

Bratman, M. E. (2009). Intention, Practical Rationality, and Self-Governance. *Ethics*, 119(April):411–443.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D., Wu, J., Winter, C., Hesse, C., Chen, M., Sigler, E., Litwin, M., Gray, S., Chess,

B., Clark, J., Berner, C., McCandlish, S., Radford, A., Sutskever, I., and Amodei, D. (2020). Language models are few-shot learners. In Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., and Lin, H., editors, *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

Byrd, D. (2019). Explaining Agent-Based Financial Market Simulation. arXivID: 1909.11650.

Byrd, D. (2021). *Responsible machine learning: Supporting privacy preservation and normative alignment with multi-agent simulation.* PhD thesis, Georgia Institute of Technology.

Byrd, D., Hybinette, M., and Balch, T. H. (2019). ABIDES: Towards High-Fidelity Market Simulation for AI Research. arXivID: 1904.12066.

Byrne, R. M. (2019). Counterfactuals in explainable artificial intelligence (XAI): Evidence from human reasoning. *IJCAI*, August:6276–6282.

Cane, P. (2019). Mens rea in tort law. *Intention in Law and Philosophy*, 20(4):129–159.

Cao, Y., Li, Y., Coleman, S., Belatreche, A., and McGinnity, T. M. (2015). Adaptive hidden Markov model with anomaly states for price manipulation detection. *IEEE Transactions on Neural Networks and Learning Systems*, 26(2):318–330.

Cartea, A., Jaimungal, S., and Wang, Y. (2020). Spoofing and Price Manipulation in Order-Driven Markets. *Applied Mathematical Finance*, 27(1-2):67–98.

CFTC (2013). Antidisruptive Practices Authority Interpretative guidance and policy statement. Technical Report RIN 3038-AD96, Commodity Futures Trading Commission.

CFTC (2018a). CFTC Finds Mizuho Bank, Ltd. Engaged

in Spoofing of Treasury Futures and Eurodollar Futures. https://www.cftc.gov/PressRoom/PressReleases/7800-18.

CFTC (2018b). CFTC Finds Mizuho Bank, Ltd. Engaged in Spoofing of Treasury Futures and Eurodollar Futures. Technical Report 7800-18, Commodity Futures Trading Commission.

CFTC (2020). CFTC Orders JPMorgan to Pay Record $920 Million for Spoofing and Manipulation. https://www.cftc.gov/PressRoom/PressReleases/8260-20.

CFTC (2021). FY2020 Division of enforcement annual report. https://www.cftc.gov/media/5321/DOE_FY2020_AnnualReport_120120/download.

Chakraborti, T., Kulkarni, A., Sreedharan, S., Smith, D. E., and Kambhampati, S. (2019). Explicability? Legibility? Predictability? Transparency? Privacy? Security? The Emerging Landscape of Interpretable Agent Behavior. *Proceedings of the Twenty-Ninth International Conference on Automated Planning and Scheduling*, page 11.

Child, J. (2017). Understanding ulterior mens REA: Future conduct intention is conditional intention. *Cambridge Law Journal*, 76(2):311–336.

Chordia, T., Roll, R., and Subrahmanyam, A. (2008). Liquidity and market efiňĄciency. *Journal of Financial Economics*, page 20.

Cliff, D. (2018). BSE: A Minimal Simulation of a Limit-Order-Book Stock Exchange. *European Modelling and Simulation Symposium (EMSS-2018)*.

Cliff, D. and Bruten, J. (1997). Minimal-Intelligence Agents for Bargaining Behaviors in Market-Based Environments. Technical report, HP Laboratories Bristol.

CNECT (2021). Proposal for a regulation of the European parliament and of the Council laying down harmonised rules on Artificial Intelligence (Artifi-

cial Intelligence Act) and amending certain union legislative acts. Technical Report COM/2021/206, European Commission, Directorate-General for Communications Networks, Content and Technology.

Coffey, G. (2009). Codifying the Meaning of 'Intention' in the Criminal Law. *The Journal of Criminal Law*, 73(5):394–413.

Cohen, P. R. and Levesque, H. J. (1990). Intention is choice with commitment. *Artificial Intelligence*, 42(2-3):213–261.

Cont, R. (2001). Empirical properties of asset returns: Stylized facts and statistical issues. *Quantitative Finance*, 1(2):223–236.

Cont, R., Kukanov, A., and Stoikov, S. (2013). The price impact of order book events. *Journal of Financial Econometrics*, 12(1):47–88.

Cooper, R., Davis, M., and Van Vliet, B. (2016). The Mysterious Ethics of High-Frequency Trading. *Business Ethics Quarterly*, 26(1):1–22.

Creswell, A. and Bharath, A. A. (2017). Denoising Adversarial Autoencoders.

Creswell, A., White, T., Dumoulin, V., Arulkumaran, K., Sengupta, B., and Bharath, A. A. (2018). Generative Adversarial Networks: An Overview. arXivID: 1710.07035.

Criminal Prosecution Service (2019). Homicide: Murder and Manslaughter. https://www.cps.gov.uk/legal-guidance/homicide-murder-and-manslaughter.

Cushman, F. (2008). Crime and punishment: Distinguishing the roles of causal and intentional analyses in moral judgment. *Cognition*, 108(2):353–380.

Dalko, V., Michael, B., and Wang, M. (2020). Spoofing: effective market power building through perception alignment. *Studies in Economics and Finance*, 37(3):497–511.

Dalko, V. and Wang, M. H. (2020). High-frequency trading: Order-based innovation or manipulation? *Journal of Banking Regulation*, 21(4):289–298.

De Cremer, D. and Kasparov, G. (2021). The ethics of technology innovation: a double-edged sword? *AI and Ethics*, pages s43681–021–00103–x.

De Graaf, M. M. and Malle, B. F. (2018). People's Judgments of Human and Robot Behaviors: A Robust Set of Behaviors and Some Discrepancies. *ACM/IEEE International Conference on Human-Robot Interaction*, pages 97–98.

De Graaf, M. M. and Malle, B. F. (2019). People's Explanations of Robot Behavior Subtly Reveal Mental State Inferences. *ACM/IEEE International Conference on Human-Robot Interaction*, 2019-March:239–248.

De Jong, F. (2011). Theorizing criminal intent: a methodological account. *Utrecht Law Review*, 7(1):1.

Dennett, D. (1987). *The Intentional Stance*. MIT Press.

Dietvorst, B. J. and Bartels, D. M. (2021). Consumers Object to Algorithms Making Morally Relevant Tradeoffs Because of Algorithms' Consequentialist Decision Strategies. *Journal of Consumer Psychology*.

Doersch, C. (2016). Tutorial on variational autoencoders. arXivID: 1606.05908v2.

Donier, J. (2012). Market Impact with Autocorrelated Order Flow under Perfect Competition. arXivID:1212.4770v1.

Duff, R. A. (1990). *Intention, Agency and Criminal Liability*. Blackwell.

Egginton, J. F., Van Ness, B. F., and Van Ness, R. A. (2016). Quote Stuffing: Quote Stuffing. *Financial Management*, 45(3):583–608.

Etzioni, A. and Etzioni, O. (2016). Designing AI systems that obey our laws and values. *Communications of the ACM*, 59(9):29–31.

Fan, R., Talavera, O., and Tran, V. (2020). Social media bots and stock markets. *European Financial Management*, 26(3):753–777.

Finn, J. K., Tregenza, T., and Norman, M. D. (2009). Defensive tool use in a coconut-carrying octopus. *Current Biology*, 19(23):1069–1070.

Fletcher, G. P. (1971). Theory of criminal negligence: A comparative analysis. *University of Pennsylvania Law Review*, 119(3):401–438.

FMSB (2018). Behavioural Cluster Analysis: Misconduct Patterns in Financial Markets. Technical report, FICC Markets Standards Board.

Franklin, M., Ashton, H., Awad, E., and Lagnado, D. (2022a). Causal framework of AI responsibility. In *Fifth AAAIACM conference on Artificial Intelligence, Ethics and Society*, Oxford, United Kingdom.

Franklin, M., Ashton, H., Gorman, R., and Armstrong, S. (2022b). Missing Mechanisms of Manipulation in the EU AI Act. *The International FLAIRS Conference Proceedings*, 35.

Franklin, M., Ashton, H., Gorman, R., and Armstrong, S. (2022c). Recognising the importance of preference change: A call for a coordinated multidisciplinary research effort in the age of AI. *The AAAI-22 Workshop on AI For Behavior Change co-located with the Thirty-Sixth AAAI Conference on Artificial Intelligence (AAAI-22)*.

Fruth, A., Schöneborn, T., and Urusov, M. (2014). Optimal trade execution and price manipulation in order books with time-varying liquidity. *Mathematical Finance*, 24(4):651–695.

Furey, J. R. (2010). *A Consistent Approach to Assessing Mens Rea in the Criminal Law of England and Wales*. PhD thesis, University of Exeter.

Furlough, C., Stokes, T., and Gillan, D. J. (2021). Attributing Blame to Robots: I. The Influence of Robot Autonomy. *Human Factors*, 63(4):592–602.

García, J. and Fernandez, F. (2015). A Comprehensive Survey on Safe Reinforcement Learning. *Journal of Machine Learning Research*, 16:1437–1480.

Gašperov, B., Begušić, S., Posedel Šimović, P., and Kostanjčar, Z. (2021). Reinforcement Learning Approaches to Optimal Market Making. *Mathematics*, 9(21):2689.

Gershman, S. J. (2015). Reinforcement learning and causal models. *Oxford Handbook of Causal Reasoning*, pages 1–32.

Ginther, M. R., Shen, F. X., Bonnie, R. J., Hoffman, M. B., Jones, O. D., Marois, R., and Simons, K. W. (2014). The language of Mens Rea. *Vanderbilt Law Review*, 67(5):1327–1372.

Gjerstad, S. (2007). The competitive market paradox. *Journal of Economic Dynamics and Control*, 31(5):1753–1780.

Gode, D. K. and Sunder, S. (1993). Allocative Efficiency of Markets with Zero-Intelligence Traders : Market as a Partial Substitute for Individual Rationality Author (s): Dhananjay K . Gode and Shyam Sunder Source : Journal of Political Economy , Vol . 101 , No . 1 (Feb ., 1993), pp . *Journal of Political Economy*, 101(1):119–137.

González-Esteban y Patrici Calvo, E. (2022). Ethically governing artificial intelligence in the field of scientific research and innovation. *Heliyon*, 8(2):e08946.

Goodfellow, I., Bengio, Y., and Courville, A. (2016). *Deep Learning*. The MIT Press.

Goodfellow, I. J., Mirza, M., Xiao, D., Courville, A., and Bengio, Y. (2015).

An Empirical Investigation of Catastrophic Forgetting in Gradient-Based Neural Networks. arXiv ID:1312.6211.

Griffith, S., Subramanian, K., Scholz, J., Isbell, C. L., and Thomaz, A. L. (2013). Policy Shaping: Integrating Human Feedback with Reinforcement Learning. *Advances in Neural Information Processing Systems (NeuIPS)*, 26.

Ha, D. and Schmidhuber, J. (2018). World Models. arXivID: 1803.10122.

Halpern, J. Y. (2016). *Actual Causality*. MIT Press.

Halpern, J. Y. and Kleiman-Weiner, M. (2018). Towards formal definitions of blameworthiness, intention, and moral responsibility. *32nd AAAI Conference on Artificial Intelligence, AAAI 2018*, pages 1853–1860.

Hart, O. D. (1977). On the Profitability of Speculation. *The Quarterly Journal of Economics*, 91(4):579–597.

Hasselt, H. V., Guez, A., and Silver, D. (2016). Double DQN.pdf. *Proceedings of the 30th AAAI Conference on Artificial Intelligence (AAAI-16)*, pages 2094–2100.

He, C., Ma, M., and Wang, P. (2020). Extract interpretability-accuracy balanced rules from artificial neural networks: A review. *Neurocomputing*, 387:346–358.

Heider, F. and Simmel, M. (1944). An Experimental Study of Apparent Behavior. *The American Journal of Psychology*, 57(2):243–259.

Hein, D., Udluft, S., and Runkler, T. A. (2018). Interpretable policies for reinforcement learning by genetic programming. *Engineering Applications of Artificial Intelligence*, 76(November 2017):158–169.

Henderson, P., Islam, R., Bachman, P., Pineau, J., Precup, D., and Meger,

D. (2018). Deep Reinforcement Learning That Matters. *Proceedings of the AAAI Conference on Artificial Intelligence*, 32(1).

Herzog, C. (2022). On the risk of confusing interpretability with explicability. *AI and Ethics*, 2(1):219–225.

Hidalgo, C. A., Orghian, D., Canals, J. A., de Almeida, F., and Martin, N. (2021). *How Humans Judge Machines*. MIT Press.

Hildebrandt, M. (2019). Closure: on ethics, code and law. In *Law for Computer Scientists*, chapter 11. Oxford University Press.

Hodge, P. (2019). The Potential and Perils of Financial Technology: Can the Law adapt to cope?

Huang, S. and Ontañón, S. (2020). A Closer Look at Invalid Action Masking in Policy Gradient Algorithms. arXivID: 2006.14171.

Isbell, C., Shelton, C. R., Kearns, M., Singh, S., and Stone, P. (2001). A social reinforcement learning agent. In *Proceedings of the fifth international conference on Autonomous agents - AGENTS '01*, pages 377–384, Montreal, Quebec, Canada. ACM Press.

Ishowo-Oloko, F., Bonnefon, J.-F., Soroye, Z., Crandall, J., Rahwan, I., and Rahwan, T. (2019). Behavioural evidence for a transparency-efficiency trade-off in human-machine cooperation. *Nature Machine Intelligence*, 1(11):517–521.

Jansen, N., K onighofer, B., Junges, S., Serban, A. C., and Bloem, R. (2020). Safe Reinforcement Learning via Probabilistic Shields. *31st international conference on concurrency theory CONCUR 2020*.

Jarrow, R. A. (1992). Market Manipulation , Bubbles , Corners , and Short Squeezes. *The Journal of Financial and Quantitative Analysis*, 27(3):311–336.

Jin, C., Allen-Zhu, Z., Bubeck, S., and Jordan, M. I. (2018). Is Q-learning provably efficient? *Advances in Neural Information Processing Systems, (NeurIPS)*, pages 4863–4873. arXivId: 1807.03765.

Johnson, D. G. and Verdicchio, M. (2019). AI, agency and responsibility: the VW fraud case and beyond. *AI and Society*, 34(3):639–647.

Jörling, M., Böhm, R., and Paluch, S. (2019). Service Robots: Drivers of Perceived Responsibility for Service Outcomes. *Journal of Service Research*, 22(4):404–420.

Kalathil, D., Borkar, V. S., and Jain, R. (2014). Empirical Q-Value Iteration.

Karpe, M., Fang, J., Ma, Z., and Wang, C. (2020). Multi-agent reinforcement learning in a realistic limit order book market simulation. In *Proceedings of the First ACM International Conference on AI in Finance*, pages 1–7, New York New York. ACM.

Kasperkevic, J. (2015). Swiss police release robot that bought ecstasy online.

Kemker, R., McClure, M., Abitino, A., Hayes, T. L., and Kanan, C. (2018). Measuring Catastrophic Forgetting in Neural Networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32 of *1*.

Kenny, A. (2013). Intention and Side Effects: the Mens Rea for Murder. In Keown, J. and George, R. P., editors, *Reason, Morality, and Law: The Philosophy of John Finnis*, chapter 7, pages 109–117. Oxford Scholarship Online.

Kingma, D. P. and Ba, J. (2014). Adam: A Method for Stochastic Optimization. arxivID: 1412.6980.

Kingma, D. P. and Welling, M. (2013). Auto-Encoding Variational Bayes. ArxivID: 1312.6114.

Kinny, D., Georgeff, M., and Rao, A. (1996). A methodology and modelling technique for systems of BDI agents. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1038:56–71.

Klass, A. B. (2007). Punitive damages and valuing harm. *Minnesota Law Review*, 92(1):83–160.

Klass, G. (2009). A conditional intent to perform. *Legal Theory*, 15(2):107–147.

Klass, G. (2012). Meaning, Purpose, and Cause in the Law of Deception. *Georgetown Law Journal*, 100:449–446.

Klass, G. and Ayres, I. (2006). New Rules for Promissory Fraud. *Arizona Law Review*, 48:957–971.

Kleiman-Weiner, M., Gerstenberg, T., Levine, S., and Tenenbaum, J. B. (2015). Inference of intention and permissibility in moral decision making. *Proceedings of the 37th Annual Conference of the Cognitive Science Society*, 1(1987):1123–1128.

Kleinberg, S. and Mishra, B. (2009). The Temporal Logic of Causal Structures. arXivID: 1205.2634.

Klöck, F., Schied, A., and Sun, Y. (2017). Price manipulation in a market impact model with dark pool. *Applied Mathematical Finance*, 24(5):417–450.

Kneer, M. (2020). Can a robot lie ? *Cognitive Science*, 45.

Kneer, M. and Bourgeois-Gironde, S. (2017). Mens rea ascription, expertise and outcome effects: Professional judges surveyed. *Cognition*, 169(August):139–146.

Knobe, J. (2003a). Intentional action and side effects in ordinary language. *Analysis*, 63:190–194.

Knobe, J. (2003b). Intentional action in folk psychology: An experimental investigation. *Philosophical Psychology*, 16(2):309–324.

Knobe, J. (2004). Intention, Intentional Action and Moral Considerations. *Analysis*, 64(2):181–187.

Knobe, J. (2006). The Concept of Intentional Action: A Case Study in the Uses of Folk Psychology. *Philosophical Studies*, 130(2):203–231.

Knobe, J. and Malle, B. (1997). The Folk Concept of Intentionality. *Journal of Experimental Social Psychology*, 33(33):101–121.

Kong, D. and Wang, M. (2014). The Manipulator's Poker: Order-Based Manipulation in the Chinese Stock Market. *Emerging Markets Finance and Trade*, 50(2):73–98.

Kramer, A. D. I., Guillory, J. E., and Hancock, J. T. (2014). Experimental evidence of massive-scale emotional contagion through social networks. *Proceedings of the National Academy of Sciences of the United States of America*, 111(29):8788–8790.

Krause, A. (2009). CS 101.2: Notes for Lecture 2 (Bandit Problems).

Lagioia, F. and Sartor, G. (2020). AI Systems Under Criminal Law: a Legal Analysis and a Regulatory Perspective. *Philosophy and Technology*, 33(3):433–465.

Lagnado, D. A. and Channon, S. (2008). Judgments of cause and blame: The effects of intentionality and foreseeability. *Cognition*, 108(3):754–770.

Law Commission (1993). *Legislating the Criminal Code: Offences against the Person and General Principles*. Number 218. HMSO.

Law Commission (The) (1989). *A criminal code for England and Wales. Volume 1: Report and draft criminal code bill*, volume 177. HMSO.

Law Commission (The) (2015a). Appendix C: Home Office Draft Bill. In *Reform of Offences The Person*, pages 212–232. William Lea Group on behalf of HMSO.

Law Commission (The) (2015b). Reform of Offences Against The Person (Report). http://www.lawcom.gov.uk/app/uploads/2015/11/51950-LC-HC555_Web.pdf.

Leangarun, T., Tangamchit, P., and Thajchayapong, S. (2019). Stock Price Manipulation Detection using Generative Adversarial Networks. *Proceedings of the 2018 IEEE Symposium Series on Computational Intelligence, SSCI 2018*, pages 2104–2111.

Lee, E. J., Eom, K. S., and Park, K. S. (2013). Microstructure-based manipulation: Strategic behavior and performance of spoofing traders. *Journal of Financial Markets*, 16(2):227–252.

Lehman, J., Clune, J., and Misevic, D. (2020). The surprising creativity of digital evolution: A collection of anecdotes from the evolutionary computation and artificial life research communities. *Artificial Life*, 26(2):274–306.

Leike, J., Martic, M., Krakovna, V., Ortega, P. A., Everitt, T., Lefrancq, A., Orseau, L., and Legg, S. (2017). AI Safety Gridworlds. arXivID: 1711.09883.

Leonard, G., Cao, Y., Haas, M., and Mocek, G. (2020). The Legal and Economic Implications from Recent UK Spoofing Cases. *Journal of Financial Compliance*, 4(2):179–193.

Liepiņa, R., Sartor, G., and Wyner, A. (2020). Arguing about causes in law: a semi-formal framework for causal arguments. *Artificial Intelligence and Law*, 28(1):69–89.

Lin, T. C. W. (2016). The new market manipulation. *Emory Law Journal*, 66:1253–1314.

List, C. and Pettit, P. (2011). *Group Agency: The Possibility, Design and Status of Corporate Agents*. Oxford Scholarship Online.

Liu, B., Polukarov, M., Ventre, C., Li, L., Kanthan, L., and Wu, F. (2022). The Spoofing Resistance of Frequent Call Markets. *AAMAS '22: Proceedings of the 21st International Confernece on Autonomous Agents and Multiagent Systems*, pages 825–832.

Lomnicka, E. (2001). Preventing and Controlling the Manipulation of Financial Markets: Towards a Definition of Market Manipulation. *Journal of Financial Crime*, 8(4):297–304.

Loveless, J. (2010). Mens Rea: Intention, Recklessness, Negligence and Gross Negligence. In *Complete Criminal Law*, chapter 3, pages 90–150. Oxford University Press, 2nd edition.

MacKenzie, D. (2022). Spoofing: Law, materiality and boundary work in futures trading. *Economy and Society*, 51(1):1–22.

Malle, B. F. and Nelson, S. E. (2003). Judging mens rea: The tension between folk concepts and legal concepts of intentionality. *Behavioral Sciences and the Law*, 21(5):563–580.

Malle, B. F., Scheutz, M., Forlizzi, J., and Voiklis, J. (2016). Which robot am I thinking about? The impact of action and appearance on people's evaluations of a moral robot. *ACM/IEEE International Conference on Human-Robot Interaction*, 2016-April(October 2017):125–132.

Mark, G. (2019). Spoofing and Layering. *Journal of Corporation Law*, 45(2):101–169.

Martínez-Miranda, E., McBurney, P., and Howard, M. J. (2016). Learning un-

fair trading: A market manipulation analysis from the reinforcement learning perspective. *Proceedings of the 2016 IEEE Conference on Evolving and Adaptive Intelligent Systems, EAIS 2016*, pages 103–109.

Maslov, S. (2000). Simple model of a limit order-driven market. *Physica A: Statistical Mechanics and its Applications*, 278(3):571–578.

Mavroudis, V. (2019). Market Manipulation as a Security Problem: Attacks and Defenses. In *Proceedings of the 12th European Workshop on Systems Security - EuroSec '19*, pages 1–6, Dresden, Germany. ACM Press.

McCloskey, M. and Cohen, N. J. (1989). Catastrophic Interference in Connectionist Networks: The Sequential Learning Problem. In *Psychology of Learning and Motivation*, volume 24, pages 109–165. Elsevier.

McIntyre, A. (2019). Doctrine of Double Effect. In Zalta, E. N., editor, *Stanford Encyclopedia of Philosophy*. Metaphysics Research Lab, Stanford University, spring 201 edition.

McManus, R. M. and Rutchick, A. M. (2019). Autonomous Vehicles and the Attribution of Moral Responsibility. *Social Psychological and Personality Science*, 10(3):345–352.

Mele, A. R. and Cushman, F. (2007). Intentional Action, Folk Judgments, and Stories: Sorting Things Out. *Midwest Studies in Philosophy*, 31(1):184–201.

Melo, F. S. (2001). Convergence of Q-learning: a simple proof. Technical report, Institute for Systems and Robotics, Lisbon, Portugal.

Mendonça, L. and De Genaro, A. (2020). Detection and analysis of occurrences of spoofing in the Brazilian capital market. *Journal of Financial Regulation and Compliance*, 28(3):369–408.

Meng, T. L. and Khushi, M. (2019). Reinforcement Learning in Financial Markets. *Data*, 4(3):110.

Mizuta, T. (2020a). An agent-based model for designing a financial market that works well. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 400–406, Canberra, ACT, Australia. IEEE.

Mizuta, T. (2020b). Can an AI perform market manipulation at its own discretion? - A genetic algorithm learns in an artificial market simulation -. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 407–412, Canberra, ACT, Australia. IEEE.

Mnih, V., Badia, A. P., Mirza, L., Graves, A., Harley, T., Lillicrap, T. P., Silver, D., and Kavukcuoglu, K. (2016a). Asynchronous methods for deep reinforcement learning. *33rd International Conference on Machine Learning, ICML 2016*, 4:2850–2869.

Mnih, V., Badia, A. P., Mirza, M., Graves, A., Lillicrap, T. P., Harley, T., Silver, D., and Kavukcuoglu, K. (2016b). Asynchronous Methods for Deep Reinforcement Learning. *arXiv:1602.01783 [cs]*. arXiv: 1602.01783.

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., Graves, A., Riedmiller, M., Fidjeland, A. K., Ostrovski, G., Petersen, S., Beattie, C., Sadik, A., Antonoglou, I., King, H., Kumaran, D., Wierstra, D., Legg, S., and Hassabis, D. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533.

Moerland, T. M., Broekens, J., and Jonker, C. M. (2017). Learning Multimodal Transition Dynamics for Model-Based Reinforcement Learning. arXivID: 1705.00470.

Molden, D. C. (2009). Finding meaning in others' intentions: The process of judging intentional behaviors and intentionality itself. *Psychological Inquiry*, 20(1):37–43.

Mueller, P. A., Solan, L. M., and Darley, J. M. (2012). When Does Knowledge

Become Intent? Perceiving the Minds of Wrongdoers. *Journal of Empirical Legal Studies*, 9(4):859–892.

Ng, A. Y., Harada, D., and Russell, S. (1999). Policy invariance under reward transformations: Theory and application to reward shaping. *ICML*, 99:279–287.

Ng, A. Y. and Russell, S. J. (2000). Algorithms for inverse reinforcement learning. *ICML*, 1.

Ormerod, D. and Laird, K. (2021a). 4. Crimes of negligence. In *Smith, Hogan, and Ormerod's Criminal Law*, pages 136–145. Oxford University Press.

Ormerod, D. and Laird, K. (2021b). 5. Crimes of strict liability. In *Smith, Hogan, and Ormerod's Criminal Law*, pages 146–179. Oxford University Press.

Öğüt, H., Mete Doğanay, M., and Aktaş, R. (2009). Detecting stock-price manipulation in an emerging market: The case of Turkey. *Expert Systems with Applications*, 36(9):11944–11949.

Owen, R. (2014). The UK Engineering and Physical Sciences Research Council's commitment to a framework for responsible innovation. *Journal of Responsible Innovation*, 1(1):113–117.

Palit, I., Phelps, S., and Ng, W. L. (2012). Can a zero-intelligence plus model explain the stylized facts of financial time series data? *11th International Conference on Autonomous Agents and Multiagent Systems 2012, AAMAS 2012: Innovative Applications Track*, 2:552–559.

Parsons, S. (2000). Intention in Criminal Law: why is it so difficult to find? *Mountbatten Journal of Legal Studies*, 4(1 & 2):5–19.

Passino, K. M., Seeley, T. D., and Visscher, P. K. (2008). Swarm cognition in honey bees. *Behavioral Ecology and Sociobiology*, 62(3):401–414.

Peng, B., Li, X., Gao, J., Liu, J., and Wong, K. F. (2018). Deep DYnA-Q: Integrating planning for task-completion dialogue policy learning. *ACL 2018 - 56th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference (Long Papers)*, 1:2182–2192.

Pettit, D. and Knobe, J. (2009). The pervasive impact of moral judgment. *Mind and Language*, 24(5):586–604.

Prakken, H. (2017). On the problem of making autonomous vehicles conform to traffic law. *Artificial Intelligence and Law*, 25(3):341–363.

Pranger, S., Könighofer, B., Tappler, M., Deixelberger, M., Jansen, N., and Bloem, R. (2020). Adaptive Shielding under Uncertainty. *American control conferences ACC*, pages 3467–3474.

Pricope, T.-V. (2021). Deep Reinforcement Learning in Quantitative Algorithmic Trading: A Review. arXivID: 2106.00123.

Quillien, T. and German, T. C. (2021). A simple definition of 'intentionally'. *Cognition*, 214(June):104806.

Raffin, A., Hill, A., Gleave, A., Kanervisto, A., Ernestus, M., and Dormann, N. (2021). Stable-Baselines3: Reliable Reinforcement Learning Implementations. *Journal of Machine Learning Research*, 22:8.

Reed, S., Zolna, K., Parisotto, E., Colmenarejo, S. G., Novikov, A., Barth-Maron, G., Gimenez, M., Sulsky, Y., Kay, J., Springenberg, J. T., Eccles, T., Bruce, J., Razavi, A., Edwards, A., Heess, N., Chen, Y., Hadsell, R., Vinyals, O., Bordbar, M., and de Freitas, N. (2022). A Generalist Agent. Technical Report arXiv:2205.06175, arXiv. arXiv:2205.06175 [cs] type: article.

Reina, A., Bose, T., Trianni, V., and Marshall, J. A. (2018). Psychophysical Laws and the Superorganism. *Scientific Reports*, 8(1):1–8.

Renault, T. (2017). Market Manipulation and Suspicious Stock Recommendations on Social Media. *SSRN Electronic Journal*.

Rezende, D. J., Mohamed, S., and Wierstra, D. (2014). Stochastic Backpropagation and Approximate Inference in Deep Generative Models.

Robbins, I. P. (1990). The Ostrich Instruction: Deliberate Ignorance as a Criminal Mens Rea. *The Journal of Criminal Law and Criminology (1973-)*, 81(2):191.

Robinson, P. H. and Darley, J. M. (1995). *Justice, Liability, and Blame : Community Views and the Criminal Law*. Faculty Scholarship at Penn Law, 1634 edition.

Ross, S. A. (2004). *No Arbitrage: The Fundamental Theorem of Finance*. Princeton Universty Press.

Russell, S. J. (2019). *Human compatible: artificial intelligence and the problem of control*. Allen Lane, an imprint of Penguin Books, London.

Sales, P. (2019). Algorithms, Artificial Intelligence and the Law. https://www.bailii.org/bailii/lecture/06.pdf.

Schrittwieser, J., Antonoglou, I., Hubert, T., Simonyan, K., Sifre, L., Schmitt, S., Guez, A., Lockhart, E., Hassabis, D., Graepel, T., Lillicrap, T., and Silver, D. (2020). Mastering Atari, Go, chess and shogi by planning with a learned model. *Nature*, 588(7839):604–609.

Schulman, J., Wolski, F., Dhariwal, P., Radford, A., and Klimov, O. (2017). Proximal Policy Optimization Algorithms. *arXiv:1707.06347 [cs]*. arXiv: 1707.06347.

Scopino, G. (2014). The (Questionable) Legality of High-Speed "Pinging" and "Front Running" in the Futures Markets. *Connecticut Law Review*, 47(3):607–697.

Scopino, G. (2015). Do Automated Trading Systems dream of manipulating the price of futures contracts? Policing markets for improper trading practices by algorithmic robots. *Florida Law Review*, 67:221.

Searle, J. R. (1999). *Mind, Language and Society: Philosophy in the real world*. Basic Books.

Sentencing Council (The) (2019). General guideline: overarching principles. https://www.sentencingcouncil.org.uk/overarching-guides/crown-court/item/general-guideline-overarching-principles/.

Shoham, Y. (1993). Agent-oriented programming. *Artificial Intelligence*, 60:51–92.

Shute, S. (2002). Knowledge and Belief in the Criminal Law. In Shute, S. and Simester, A., editors, *Criminal Law Theory: Doctrines of the General Part*, chapter 8. Oxford Scholarship Online.

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., Chen, Y., Lillicrap, T., Hui, F., Sifre, L., Van Den Driessche, G., Graepel, T., and Hassabis, D. (2017). Mastering the game of Go without human knowledge. *Nature*, 550(7676):354–359.

Simester, A. P. (2021). *Fundamentals of criminal law: responsibility, culpability, and wrongdoing*. Oxford University Press, Oxford, United Kingdom, first edition edition. OCLC: on1242932280.

Simester, A. P., Spencer, J. R., Stark, F., Sullivan, G. R., and Virgo, G. J. (2019). Mens Rea. In *Simester and Sullivan's Criminal Law*, chapter 5, pages 137–190. Hart, 7 edition.

Smith, J. C. (1990). A note on " intention ". *Criminal Law review*, 85:85–91.

Smith, V. L. (1993). When prior knowledge and law collide - Helping jurors use the law. *Law and Human Behavior*, 17(5):507–536.

Sohn, K., Lee, H., and Yan, X. (2015). Learning Structured Output Representation using Deep Conditional Generative Models. In *Advances in Neural Information Processing Systems*, pages 3483–3491.

Sommers, R. (2021). Experimental Jurisprudence. *Science*, 373(6553).

Stark, F. (2017). Introduction. In *Culpable Carelessness: Recklessness and Negligence in the Criminal Law*, chapter 1, pages 1–25. Cambridge University Press.

Stenfors, A., Doraghi, M., Soviany, C., Susai, M., and Vakili, K. (2022). Cross-Market Spoofing. Technical report, University of Portsmouth.

Stenfors, A. and Susai, M. (2021). Spoofing and pinging in foreign exchange markets. *Journal of International Financial Markets, Institutions and Money*, 70:101278.

Stevens, R. and Zhang, J. Y. (2016). Slipping through the Cracks: Detecting Manipulation in Regional Commodity Markets. *SSRN Electronic Journal*.

Storey, T. (2019). Inchoate Offences. In *Unlocking Criminal Law*, chapter 6, pages 137–170. Routledge, 7th edition.

Sutton, R. S. (1990). Integrated architectures for learning, planning, and reacting based on approximating dynamic programming. In Proceedings of the 7th. *International Conference on Machine Learning*, pages(1987):216–224.

Sutton, R. S. and Barto, A. G. (1998). *Reinforcement learning: an introduction*. Adaptive computation and machine learning. MIT Press, Cambridge, Mass.

Sutton, R. S. and Barto, A. G. (2018). *Reinforcement Learning: An Introduction*. MIT Press, 2nd edition.

Sutton, R. S., Szepesvári, C., Geramifard, A., and Bowling, M. (2008). Dyna-style planning with linear function approximation and prioritized sweeping. *Proceedings of the 24th Conference on Uncertainty in Artificial Intelligence, UAI 2008*, pages 528–536.

Tao, X., Day, A., Ling, L., and Drapeau, S. (2022). On detecting spoofing strategies in high-frequency trading. *Quantitative Finance*, pages 1–21.

Taylor, G. (2004). Concepts of Intention in German Criminal Law. *Oxford Journal of Legal Studies*, 24(1):99–127.

The American Law Insitute (2017). General Requirements of Culpability.

The Law Commission (2007). Conspiracy and Attempts: A consultation paper.

Thellman, S., Silvervarg, A., and Ziemke, T. (2017). Folk-psychological interpretation of human vs. humanoid robot behavior: Exploring the intentional stance toward robots. *Frontiers in Psychology*, 8(NOV):1–14.

Tobia, K. (2021). Law and the Cognitive Science of Ordinary Concepts. In *Law and Mind*, pages 86–96. Cambridge University Press.

Tobia, K. (2022). Experimental Jurisprudence. *University of Chicago Law Review*, 89.

Van Hasselt, H. (2010). Double Q-learning. *Advances in Neural Information Processing Systems 23: 24th Annual Conference on Neural Information Processing Systems 2010, NIPS 2010*, pages 1–9.

van Kervel, V. (2015). Competition for Order Flow with Fast and Slow Traders. *The Review of Financial Studies*, 28(7):2094–2127.

Verma, A., Murali, V., Singh, R., Kohli, P., and Chaudhuri, S. (2018). Pro-

grammatically interpretable reinforcement learning. *35th International Conference on Machine Learning, ICML 2018*, 11:8024–8033.

Verma, I. M. (2014). Editorial expression of concern: Experimental evidence of massive-scale emotional contagion through social networks. *Proceedings of the National Academy of Sciences of the United States of America*, 111(29):10779.

Wah, E., Hurd, D., and Wellman, M. (2015). Strategic Market Choice: Frequent Call Markets vs. Continuous Double Auctions for Fast and Slow Traders. In *Proceedings of the The Third Conference on Auctions, Market Mechanisms and Their Applications*, Chicago, United States. ACM.

Walker, J., Doersch, C., Gupta, A., and Hebert, M. (2016). An uncertain future: Forecasting from static images using variational autoencoders. *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 9911 LNCS:835–851.

Wang, Q., Xu, W., Huang, X., and Yang, K. (2019). Enhancing Intraday Stock Price Manipulation Detection by Leveraging Recurrent Neural Networks with Ensemble Learning. *Neurocomputing*, 347:46–58.

Wang, X. (2021). *Computational Modeling and Design of Financial Markets: Towards Manipulation-Resistant and Expressive Markets*. PhD thesis, University of Michigan.

Wang, X., Hoang, C., and Wellman, M. P. (2020). Learning-based trading strategies in the face of market manipulation. In *Proceedings of the First ACM International Conference on AI in Finance*, pages 1–8, New York New York. ACM.

Wang, X., Vorobeychik, Y., and Wellman, M. P. (2018). A cloaking mechanism to mitigate market manipulation. *IJCAI International Joint Conference on Artificial Intelligence*, 2018-July:541–547.

Wang, X. and Wellman, M. P. (2017). Spoofing the Limit Order Book: An Agent-Based Model. *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems. International Foundation for Autonomous Agents and Multiagent Systems*, pages 651–659.

Weir, A. A., Chappell, J., and Kacelnik, A. (2002). Shaping of hooks in new caledonian crows. *Science*, 297(5583):981.

Wen, M. and Topcu, U. (2018). Constrained Cross-Entropy Method for Safe Reinforcement Learning. *Advances in Neural Information Processing Systems*, 31:7461–7471.

Williams, G. (1987). Oblique intention. *The Cambridge Law Journal*, 46(3):417–438.

Williams, R. J. (1992). Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256.

Winfield, A., Blum, C., and Liu, W. (2014). Towards an Ethical Robot; Internal Models, Consequences and Ethical Action Selection. In Mistry, M., Leonardis, A., Witkowski, M., and Melhuish, C., editors, *Advances in Autonomous Robotics Systems: 15th Annual Conference, TAROS 2014, Birmingham, UK, September 1-3, 2014. Proceedings*, volume 8717 of *Lecture Notes in Computer Science*, pages 85–96. Springer International Publishing.

Withanawasam, R. M., Whigham, P. A., Crack, T., and Premachandra, I. M. (2010). An empirical investigation of the Maslov limit order market model. *The Information Science Discussion Paper Series*, 4.

Withanawasam, R. M., Whigham, P. A., and Crack, T. F. (2013). Characterising trader manipulation in a limit-order driven market. *Mathematics and Computers in Simulation*, 93:43–52.

Yaffe, G. (2004). Conditional intent and mens rea. *Legal Theory*, 10(4):273–

310.

Yagemann, C., Chung, S. P., Uzun, E., Ragam, S., Saltaformaggio, B., and Lee, W. (2020). On the Feasibility of Automating Stock Market Manipulation. In *Annual Computer Security Applications Conference*, pages 277–290, Austin USA. ACM.

Yavar Bathaee (2018). The artificial intelligence black box and the failure of intent and causation. *Harvard Journal of Law & Technology*, 31(2):890–938.

Yeo, N. (2020). Mistakes and knowledge in algorithmic trading : the Singapore Court of Appeal case of Quoine v B2C2. *Journal of International Banking and Financial Law*, 35(5):300–305.

Young, L. and Saxe, R. (2011). When ignorance is no excuse: Different roles for intent across moral domains. *Cognition*, 120(2):202–214.

Zhang, Y., Tino, P., Leonardis, A., and Tang, K. (2021). A Survey on Neural Network Interpretability. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 5(5):726–742.

Zulkifley, M. A., Abd Sukor, M. E., Munir, A. F., and Mohd Shafiai, M. H. (2021). Stock Market Manipulation Detection using Artificial Intelligence: A Concise Review. In *2021 International Conference on Decision Aid Sciences and Application (DASA)*, pages 165–169, Sakheer, Bahrain. IEEE.