Human, Artificial Intelligence and Tail Risk

黄仕捷
Shijie Huang

A dissertation submitted
in partial fulfillment of
the requirements for the degree of
Doctor of Philosophy
in Decision, Risk and Financial Sciences

Committee in charge:

Professor Peter Bossaerts
Professor Carsten Murawski
Professor Shinsuke Suzuki
Assistant Professor Nitin Yadav

Centre for Brain, Mind and Markets
Faculty of Business and Economics
University of Melbourne

March 2023

COPYRIGHT

Title: Human, Artificial Intelligence and Tail Risk.

This version: March 2023. Click here for the latest version.

 0000-0002-3641-6169.

Human, Artificial Intelligence and Tail Risk

Shijie Huang

Abstract

I study learning efficiency of artificial and human agents in the environment of financial markets. One prominent feature that distinguishes this environment is tail risk, which means that outliers are more frequent and substantial relative to Gaussian outliers. Failure to account for tail risk deteriorates learning efficiency, causing agents to derail from optimal actions. In the dissertation, I explore improvements to learning by artificial agents under tail risk, and whether human learning exhibits similar improvements. Finally I study to what extent agents' interactions and intelligence level would cause or amplify tail risk.

A key success of artificial intelligence has been reinforcement learning. I first show that even the most advanced reinforcement learning protocol yields sub-optimal behavior in an environment with tail risk. Inspired by the concept of statistical efficiency, I propose a solution that nicely complements a recent protocol – distributional reinforcement learning – and improves the performance of algorithms. I show that the proposed algorithm learns much faster and is robust once it settles on a policy.

Thus, efficiency gains are possible for artificial agents. Do humans exhibit the same kind of adjustment in an environment of tail risk? In the second study, I design an experiment to examine whether and how efficiency concerns drive human learning of stochastic rewards. While I find substantial heterogeneity, overall the answer is affirmative. Efficiency gains translate into enhanced choice confidence, except when participants fail to discover the most efficient estimator.

In finance, the real causes of tail risk remain elusive. One conjecture is that, even without triggers from any extreme event, tail risk emerges because of agents' interactions in the marketplace. Motivated by the zero-intelligence and machine learning literature, I propose a paradigm to approach this conjecture in the third study. The paradigm comprises a single-widget economy, a continuous open-book market, and a group of trading agents with different intelligence levels. I demon-

strate that trading generates excessive tail risk even when the underlying economic shifts follow a Gaussian law. Introducing a profit-seeking market maker further increases leptokurtosis, but the tail risk is not worsened. The latter suggests that tail risk and leptokurtosis may need to be distinguished.

致我的父母：范文、黄敏华

To my parents Minhua Huang & Wen Fan

致我的妻子：曹馨如

To my wife Xinru Cao

DECLARATION

I declare that:

- the thesis comprises only my original work except where indicated in the Preface.
- due acknowledgement has been made in the text to all other materials used.
- the thesis is fewer than $100,000$ words in length, exclusive of tables, maps, bibliographies and appendices.

.....

Signature

PREFACE

The paper-based dissertation presented here covers interdisciplinary research topics. Centering around *tail risk*, it encompasses discipline insights from computer science, human behavioral studies, and research in economics & finance.

While I always yearn for a monograph of individual agents' behavior in different environments, one predicament for interdisciplinary research is that it is difficult for someone to apprehend knowledge or frameworks that they are unfamiliar with. Academic reviews and comments on cross-disciplinary topics can often be unduly scornful or even scathing. Hence, while I do not ask for indulgence for mistakes that are obvious and incontestable, I will be very grateful if readers could objectively and magnanimously judge this work.

The technicalities are regrettably inevitable, but I strive to unveil obfuscations and discipline-specific terminologies in plain language for a broader audience. Thus, I do apologize if readers with encyclopedic knowledge ever find me paraphrasing terminologies that are seemingly straightforward. I sincerely hope that this dissertation is not perceived to be dreadfully dull.

Credits

Chapter 4 is published in the following journal article. Chapter 5 and Chapter 6 are unpublished materials.

- Bossaerts, P., Huang, S., & Yadav, N. (2020). Exploiting distributional temporal difference learning to deal with tail risk. Risks, 8(4), 113.

 Queries from participants at the 2019 Multi-disciplinary Conference on Reinforcement Learning and Decision Making in Montreal, Canada (RDLM 2019), especially from Peter

Dayan, are gratefully acknowledged. We also thank Elena Asparouhova and Jan Nielsen for comments on a preliminary version of this paper. We also benefited from comments from three anonymous reviewers.

Contribution declaration: S.H. contributed 80% of the content of the article. Conceptualization, methodology and validation, P.B., S.H. and N.Y.; software, S.H. and N.Y.; formal analysis, P.B.; investigation, resources and data curation, S.H.; writing-original draft preparation, S.H.; writing-review and editing, P.B., S.H. and N.Y.; visualization, S.H.

Dissertation review

I would like to thank two anonymous reviewers for their time reviewing this dissertation and providing constructive feedback.

Funding

I am grateful and honored to have received a stipend from the Australian Government Research Training Program (RTP) Scholarship. I am also supported by the Faculty of Business and Economics Research Abroad Travelling Scholarship.

Acknowledgement of Country

I would like to acknowledge the Wurundjeri people who are the Traditional Custodians of this Land on which I completed this research. I would also like to pay respect to the Elders both past and present of the Kulin Nation and extend that respect to other Indigenous Australians present.

ACKNOWLEDGEMENTS

It has been a great pleasure to pursue a research career for the past six years down under. What an incredible adventure! Admittedly, doing a PhD was both enjoyable and arduous, but nonetheless I was fortunate enough to have a group of people whom I enjoyed learning from and talking to. In the past six years of pursuing a research career, I have accumulated many debts to all of you. I will cherish our relationship, your help, and your support for the foreseeable future.

Supervisors and Mentors

First and foremost, I am sincerely grateful to all my supervisors, Professor Peter Bossaerts, Assistant Professor Nitin Yadav, Professor Shinsuke Suzuki (鈴木真介), and Professor Carsten Murawski, for your endless guidance and magnanimity to my grievances, for providing me with great opportunities to learn and work on exciting topics, and for your encouragement blended with structural advice since my honors and throughout my entire PhD.

Pursuing a research degree during the Covid pandemic was particularly awful. However, I was tremendously fortunate to receive continuous support from all my supervisors throughout the pandemic. The wealth of knowledge and skills I have learned from them is, without doubt, a life-long treasure.

Family

Above all, I must send my special thanks to my family members; my parents Minhua Huang and Wen Fan, who gave birth to me and raised me for twenty years; and my wife Xinru Cao, whom I happily married during my PhD. Their steadfast support is the most important reason I can survive a PhD during the covid pandemic. I

have drawn great comfort from their warmth and affection, and I wish to express my sincere and irrepressible gratitude to them.

Lab Members and Friends

The Centre for Brain, Mind and Markets (formerly known as the Brain, Mind and Markets Lab) holds its unique place in my heart. Lab members are just awesome.

I want to particularly thank my PhD cohort peers, Xiaping Lu and Jeremy Metha. We made it through some challenging PhD level subjects. Together we ventured through the myth of being the very first master-PhD cohort, as well as the beacon for the future cohorts. As young scholars, their intellectual vitality in research is what I admire the most.

A special thank you should also be given to lab members from whom I received enormous support. In particular, Assistant Professor Felix Fattinger, who shaped my teaching skills and taught me economics & asset pricing; and Doctor Juan Pablo Franco, for running the very first faculty wise PhD student society together; and Abhijeet Anand, our technical genius and camping expert, for his solid knowledge of engineering and coding. His expertise partially shaped my coding habits and project repository management; these skills are considered critical for a successful PhD; and Doctor Elizabeth Bowman, for sorting out tedious ethics applications and administrative issues; and last but not least, Associate Professor Kristian Rotaru from Monash University, for very fruitful and interesting market experiments that I was involved with in his lab.

I would also like to thank peers in my honors cohort (honors in finance 2016), particularly Shireen Tang, Yang (Edison) Cheng, Bosco Feng, Trang Tran, Xinyi (Cora) Shi, Jasmine Zheng, Xueqing (Eva) Chen, Yunxiang (Richard) Geng, Edouard Lyndt; and thank you to later honors cohorts for having interesting research projects and experiments with me. It has been a great pleasure to have all of you as peers during my honors and PhD.

- 2017: Tingxuan (Charlie) Wang, Dan Wu, Jacky Kuang, Eileen Wang, Frans Van Den Bogaerde, Karlo Doroc.

- 2018: Katherine Xu, Michelle Lee, Anthony Hsu, Anirudh Suthakar, Yajie (Katrina) Wu.
- 2019: Xianghui (Renee) He, Xinan (Alex) Fan, Max Hunt, Max Ruan, Shirley Tang, Tony Wu, Michael Tirtana and Terry Wang.

I would like to thank visiting scholars from all over the world. All of you have enriched my knowledge across different fields. In particular, I would like to thank two peer researchers. Firstly, I had a very enjoyable time discussing exciting research topics in finance with Doctor Jonathan Krakow from the University of Zurich. Secondly, I would also like to thank Matthew Farrugia Roberts, with whom I had the greatest fun discussing distributional reinforcement learning. Matthew partially inspired much of the discussions on distributional reinforcement learning in this dissertation.

I was fortunate to have visited the BrainPark at Monash University in 2019, where Doctor Xiaoliu Zhang hosted me for one semester. I sincerely appreciate the visiting opportunity and her hospitality.

Finally, I would like to thank my tennis coach Junaid Hossian and my tennis mates Lee Tay, Mike Xie and Bosco Feng, for keeping me physically fit; and all my friends for keeping me mentally healthy by checking in with me on a regular basis.

Resource

I appreciate the Melbourne Research Cloud team for providing me with a dedicated cloud server, which allows me to run simulations and host online experiments, particularly during Covid-19 lockdowns. I am also grateful to the open-source community, institutions, and companies for providing excellent software and packages to students without charge; a list of acknowledgements can be found in Appendix A.3.

Yours sincerely
Harvey Huang
Melbourne, Australia
March 2023

CONTENTS

LIST OF FIGURES

LIST OF TABLES

COPYRIGHT MATERIALS

- Chapter 4, copyright permission obtained from the co-authors and from the publisher under the Creative Commons Attribution 4.0 International (CC BY 4.0).
- Futuristic droid robot in Chapter 1, copyright permission obtained from Adobe image stock under the Standard License.
- Figure 1.1 Examples of super-human performance AI, copyright permission obtained under the Creative Commons Attribution 4.0 International (CC BY 4.0).
- Figure 1.4 Example of a multi-armed bandit, copyright permission obtained under the Apache License 2.0.
- Figure 2.2 Agent-environment interaction, copyright permission obtained from Adobe image stock under the Standard License.
- Figure 3.1 Taxonomy of reinforcement learning algorithms, copyright permission obtained from OpenAI introduction of deep RL under the MIT License.

Part I

INTRODUCTION

1

INTRODUCTION AND MOTIVATION

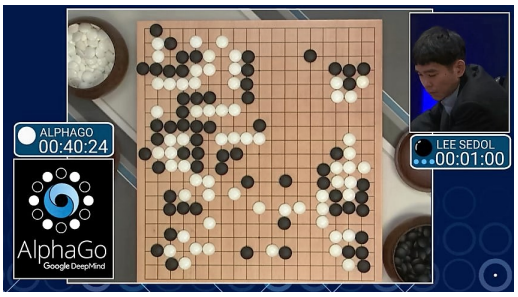
- Human and Artificial Intelligent Agents[1]

Endowing machines with abilities to perform various tasks autonomously and efficiently akin to that of humans in different environments is one of the most challenging tasks in artificial intelligence (AI) research (Finn, 2018; Lee, Sungik, & Chung, 2019; Ye, Liu, Kurutach, Abbeel, & Gao, 2021). Hitherto the most advanced AIs, such as Alpha-Go and its successors, have outperformed human experts in gaming settings (see Figure 1.1) (Mnih et al., 2015; Silver et al., 2016, 2017; Moravčík et al., 2017; N. Brown & Sandholm, 2019; Vinyals et al., 2019). Despite the super-human performance, the efficiency of these state-of-the-art AI agents remains abysmal. For example, AlphaGo Zero requires asynchronous training with 4.9 million matches over 72 hours (Silver et al., 2017); the number of training matches is an order of magnitude larger than what a human grand-master can play in a lifetime[2]. Thus,

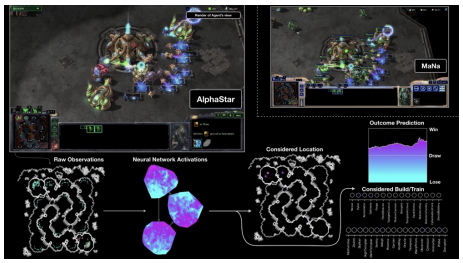
[1]Source: Futuristic droid robot, Adobe image stock under the Standard License.
[2]Assuming five matches a day for 90 years, a human expert can play at maximum 5 * 365 * 90 = 165,250 matches in a lifetime.

an emerging ramification in AI research aims to design sample efficient algorithms to perform well under various environments but with a lower number of training rounds (Yu, 2018; Kapturowski et al., 2022).

(a) Atari® games (Mnih et al., 2015)



(b) Game of Go (Silver et al., 2016)



(c) RTS games (Vinyals et al., 2019)

(d) Poker (N. Brown & Sandholm, 2019)

Figure 1.1. Examples of super-human performance AI[3]. RTS games: real-time strategy games. Here the example of an RTS game is Starcraft II®.

Meanwhile, the proliferation of machine learning (ML) and AI has spanned financial markets. Both the financial industry and academics have expressed growing interest in applying ML and AI in asset pricing, trading, market making, portfolio management, etc. The focus is slightly different between industry and academia but shares similar traits in a broad scope. Industry researches utilize ML and AI for prediction and objective optimization, e.g., Ganesh et al. (2019); Karpe, Fang, Ma, and Wang (2020); Amrouni, Moulin, and Balch (2022) whereas academics use ML algorithms as tools alternative to econometric models for research topics like asset pricing (Gu, Kelly, & Xiu, 2020; Bianchi, Büchner, & Tamoni, 2021; Leippold,

[3]Source: (a) https://atari.com/collections/games (b) https://www.youtube.com/watch?v=vFr3K2DORc8 (c) https://www.youtube.com/watch?v=HcZ48JDamyk&t=282s (d) https://www.pexels.com/photo/cards-casino-chance-chip-269630/.

Wang, & Zhou, 2022), corporate finance (N. Cohen, Balch, & Veloso, 2020) and statistical inference (Farrell, Liang, & Misra, 2021).

Financial markets are often regarded as "games" among market participants, e.g., regulators, market makers, human and algorithmic traders. While there is no doubt that state-of-the-art AI agents can outperform humans in a variety of strategic games, to date, there exists little evidence that AI agents can systematically outperform human traders or non-intelligent agents in the context of financial markets[4]. One possible obstacle, I conjecture, is that financial markets are known to be inherently different from typical video games due to their unique properties of uncertainty. One crucial difference is *tail risk*: feedbacks from this environment to agents[5] contain extreme but frequent shocks. The price of typical security can vary either in small amounts or frequently experience large "jumps" in either direction. As a result, agents may perceive that markets are quite "chaotic". Figure 1.2 depicts an example of the price movement of typical security.

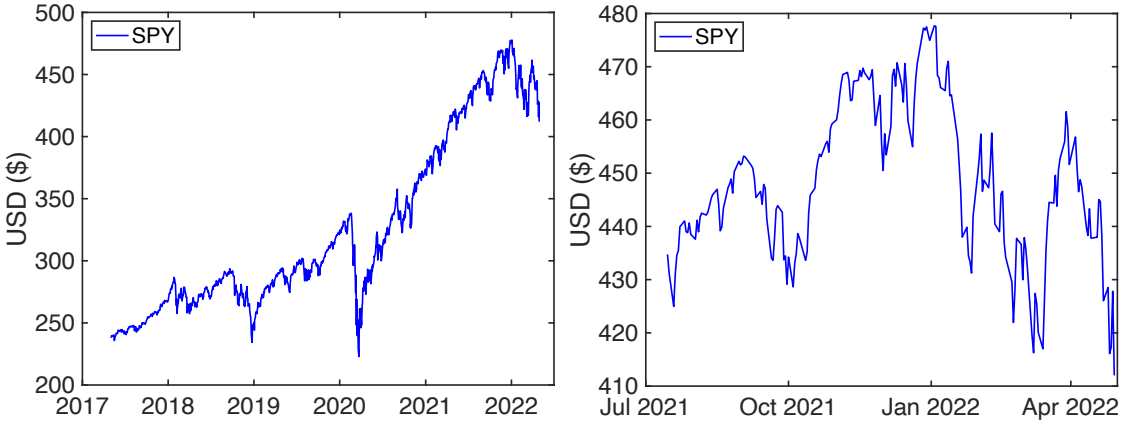


Figure 1.2. Time series data of daily SPY prices from 2017-2022 (left) and from July 2021 to April 2022 (right). SPY is an exchange-traded fund on the S&P500 index.

[4]Except for sub-second trading tasks, such as high-frequency trading.
[5]Throughout this thesis, I use the term *agents* to refer to both human and artificial agents.

Statistically, the probability distribution of daily return[6] (reward) for a typical security exhibits *leptokurtosis* (see Figure 1.3): a clustered mean and heavy tails. The return distribution has more observations within ±1 standard deviation than the Gaussian distribution, and the density in both tails of the distribution is larger than the Gaussian distribution; the kurtosis of the distribution (the fourth central moment) is much larger than 3 (d'Acremont & Bossaerts, 2016; Taleb, 2020). In a nutshell, this distinguishing statistical property indicates that the tails' outliers are disproportionally more frequent and substantial relative to Gaussian outliers, hence the name "tail risk" (Mandelbrot, 1963).

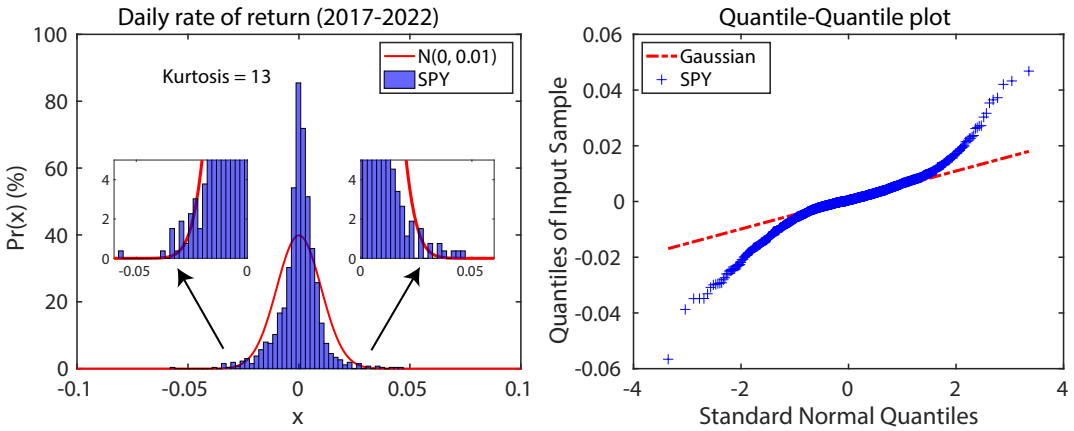


Figure 1.3. Empirical probability distribution (left) and Quantile-Quantile plot (right) of daily price change from July 2017 to April 2022. Varying the dates and years will slightly change the number (e.g., kurtosis) but does not alter the conclusion drawn by the author.

Contradicting the common perception that tail risk events are of black swan type, they occur far more frequent than black swan events (Taleb, 2007, 2020). To exemplify the heavy-tailedness nature of reward feedback from financial markets to its agents, let us consider the probability of a tailed-risk event in different scenarios. The theoretical probability of drawing a five-sigma event from a Gaussian

[6]The term "return" refers to percentage investment gain/loss obtained over a certain time frame (e.g., daily). It should not be confused with the term "return" referred to in the RL literature. However, the two terms are related as they both refer to some feedback from the environment/market to agents/investors. Likewise, we should be aware of the context (finance/RL) for proper interpretation of the terms "payoffs" or "rewards."

distribution[7] is around 3.5×10^{-7}, that is, one in 3.5 million draws. Conversely, the global financial markets experienced five-sigma events eight times in March 2020 alone upon the onset of the Covid pandemic[8]. Thus, tail risk in financial markets is not only salient but also frequent (Mandelbrot, 1963; Barro, 2006; d'Acremont & Bossaerts, 2016).

Tail risk does not only occur in financial markets, the phenomenon also exists in natural or human disaster. For instance, earthquakes are heavy-tailed (Pisarenko & Rodkin, 2010); contagious disease (like Covid-19) are strongly heavy-tailed (Cirillo & Taleb, 2020); casualties, as a consequence of armed conflicts, are also heavy-tailed (Clauset & Woodard, 2013; J. A. Friedman, 2015; Cirillo & Taleb, 2016). Hence, research on tail risk is a desideratum for the society due to the severe consequences and high frequency of disasters.

Tail risk has been well documented in statistics and financial econometrics literature. Finance literature considered tail risk as one of the critical features that explain asset returns, and risk exposures, e.g., risk premium (Kelly & Jiang, 2014), or downside investment beta (Ang, Chen, & Xing, 2006). Conventional models of the environment are developed for risk measurement, asset pricing, and prediction purpose; popular paradigms to model tail risk include, inter alia, value at risk and expected shortfall (Markowitz, 1952; Roy, 1952; Duffie & Pan, 1997), extreme value theorem (Beirlant, Goegebeur, Segers, & Teugels, 2004; De Haan & Ferreira, 2006; Cirillo & Taleb, 2020), GARCH models (Orlowski, 2012) and power-law distributions (Gabaix, Gopikrishnan, Plerou, & Stanley, 2003; Clauset, Shalizi, & Newman, 2009; Kelly & Jiang, 2014; J. A. Friedman, 2015). From a machine learning perspective, those models are effectively environmental models where (mostly) parametric views of the environment are formed; central to the goal of the conventional tail risk literature is to find a model that best describes the environment.

While a plethora of efforts has been devoted to finding an optimal model to describe the environment, how agents behave and learn in an environment with tail risk has received relatively limited attention. Some literature has documented that downside risk and market turmoil ("rare" disasters) shape retail investors' risk pref-

[7] For example, draw 0.05 from $N(0, 1)$.

[8] https://www.forbes.com/sites/lizfrazierpeck/2021/02/11/the-coronavirus-crash-of-2020-and-the-investing-lesson-it-taught-us/?sh=384b5a0e46cf

erences and decision-making in a probability setting, e.g., Rietz (1988); Barro (2006); Gabaix (2012). More recently, by virtue of the increasing retail investor presence in the financial markets and more "frequent" tailed events in both traditional markets and cryptocurrency markets, we have seen a growing number of interests on individual investor behavior and risk preferences, e.g., Chapkovski, Khapko, and Zoican (2021); Kalda, Loos, Previtero, and Hackethal (2021); Barber, Huang, Odean, and Schwarz (2022).

Indeed, many events in our daily life have consequences that are normally distributed, and thus human experience and intuitions are typically shaped around the normal realm. However, such intuition or cognitive biases could lead to inefficient estimations, or even ghastly disastrous decisions; thus it prompts questions on agents' learning and behavior in heavy-tailed environments. Can state-of-the-art AI agents or human handle tail risk? Will they survive crippling market avalanches like the ones in January 2008 or March 2020? How efficient are their prediction and estimations? Can they learn optimal actions? Does their interaction with the system enhance or even lead to tail risk?

With the prevalence of AI in financial markets, ignoring the presence of leptokurtosis and tail risk can be vicious to both markets and their participants. In an environment with tail risk, estimation efficiency becomes immensely critical for agents to figure out the best action to perform quickly. Sample average, a method that underpins many AI algorithms and animal behavior models, is no longer the most statistically efficient method to estimate the expected value of a heavy-tailed distribution (Casella & Berger, 2021). Consequently, failure to account for tail risk during learning and decision-making can severely deteriorate learning success and efficiency, causing agents to derail from optimal actions. Moreover, the repercussion of frequent and large shocks on agents endures, and taking stock of agents' behavior could help us unveil the cause of leptokurtosis. Thus, questions on agents' behavior and learning efficiency in such an environment, like the aforementioned ones, merit more perusal.

While stochasticity and uncertainty show up in several aspects of the environment, this dissertation concerns reward uncertainty[9]. Prior research on agents' behavior under reward uncertainty (or decision-making on risky choices) constitutes a scheme of cross-disciplinary topics in the field of economics, computer science, and animal behavioral research. Much behavioral research on reward uncertainty is framed as *stochastic bandit tasks*; that is, agents choose among options, typically by clicking buttons, in which rewards are drawn randomly from some probability distribution. The goal is to *learn*, through sampling, which option (bandit's arm) gives the highest expected reward on average. See Figure 1.4 for an example multi-armed bandit task.

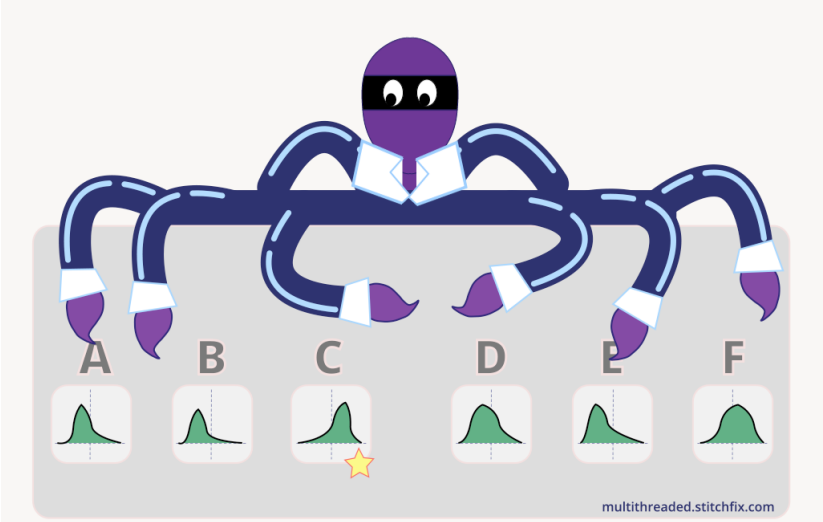


Figure 1.4. An example of k-armed ($k = 6$) stochastic bandit task. The agent is asked to learn which arm has the highest expected rewards through sampling. Source: `https://github.com/stitchfix/mab`. Reprinted with permission under the Apache License 2.0.

Research on reward uncertainty can be further divided into two different dimensions. Firstly, the reward distribution of a bandit's arm can be either stationary or non-stationary to reflect "*contingency switch*" (Dayan & Yu, 2002; Behrens, Woolrich, Walton, & Rushworth, 2007; Payzan-LeNestour & Bossaerts, 2011; Payzan-

[9]There are other sources of uncertainty, such as state transition or computational complexity, see Bossaerts and Murawski (2017); Bossaerts, Yadav, and Murawski (2019).

LeNestour, Dunne, Bossaerts, & O'Doherty, 2013), i.e., optimal actions could switch over time as a result of a state regime shift. The type of bandit task with a non-stationary reward function is often referred to as "restless bandit", a task which is often utilized by scholars who are interested in *exploration-exploitation* dilemma (Bornstein, Khaw, Shohamy, & Daw, 2017; Navarro, Tran, & Baz, 2018). The second dimension concerns whether the uncertainty is *expected* or not (Dayan & Yu, 2002; Bossaerts, 2022). When agents are informed about possible contingency shifts, uncertainty is still expected. However, when agents are not informed, learning becomes egregiously difficult since agents are required to conduct two processes of estimation and learning simultaneously: (1) the expected reward of each arm and (2) whether the regime has shifted.

This dissertation investigates another dimension that concerns the domain of reward distributions where sample average is no longer the most efficient estimation of the expected reward. One insidious limit in existing literature, among others[10], is that the reward distribution of most stochastic bandit tasks follows either binomial or Gaussian law. While this assumption provides more flexibility in computational modeling agents' choices, it may promote undue consensus since this entrenched assumption fails to capture the rich and complicated nature of real-world feedback (rewards) (Silver, Singh, Precup, & Sutton, 2021). The majority of the literature have been equivocal about whether modeling results are bounded by this distribution assumption, or whether conclusions are robust to other ambient conditions, particularly to reward functions; yet evidently agents' behavior and learning efficiency can be twisted upon receiving heavy-tailed rewards.

Some prior literature have studied humans learning on *expected* heavy-tailed rewards cases using tasks alternative to the bandit setting; it has been empirically and experimentally documented that humans often overreact to tail risk but learn to become very efficient (Barberis, 2013; d'Acremont & Bossaerts, 2016). However, convention behavioral models are computationally too complex (Bayesian) or

[10]For instance, do agents indeed behave as if they continually optimize? Research objectives on agents' behavior in stochastic bandit tasks vary but share similar traits between disciplines. In computer science, the aim is to design algorithms that yield optimal solutions with better efficiency. On the other hand, behavioral economics and animal behavioral research attempt to rationalize agents' decision-making process, unveiling models that justify agents' choices. In virtually all cases, agents' choices are either designed to optimize or assumed "as if" they are optimal (Bossaerts, 2022).

too slow to match human performance (model-free RL) (d'Acremont & Bossaerts, 2016; Bossaerts, 2018, 2022). It seems difficult to decipher the rationale behind human-level efficiency on expected heavy-tailed rewards. Thus, by relaxing the Gaussian/uniform reward assumption in a stochastic bandit task framework, I seek to systematically understand agents' learning efficiency on expected rewards. I hypothesize that the essence of unveiling the rationale behind human efficiency and potentially helping improve AI efficiency is the concept of *statistical efficiency* (Casella & Berger, 2021). The two main research questions:

1. *Can state-of-the-art artificial intelligent agents learn optimal actions efficiently when the stochastic bandit task entails heavy-tailed rewards?*
2. *How efficient are agents' learning and estimation in a stochastic bandit task where the reward distributions are non-Gaussian/binomial?*

In addressing the above two questions, I assume that the environment entails heavy-tailed rewards by default; the narrative leans more towards *agents vs. environment*. A more prominent question in finance worth pondering is where does leptokurtic return distribution come from? Current consensus appears to believe that leptokurtosis arises from tail events, but is it possible that even without triggers from any extreme event, tail risk emerges purely because of agents' interactions in the marketplace[11]? To date, there is no compelling evidence to rule out this conjecture, and there is lack of undimmed academic research paradigms on the true cause of leptokurtosis (Farmer & Lillo, 2004). Thus, the third principal research question of this dissertation is:

3. *To what extent is leptokurtosis solely the consequence of the interaction of agents through standard market microstructure, namely, the continuous open-book system?*

Motivated by the zero-intelligence and machine learning literature, I propose a paradigm to study this question (Gode & Sunder, 1993; Smith, 1962). The paradigm comprises a single-widget economy, a continuous open-book market, and a group of rational agents who interact with each other in the market through trading. In order to control for tail-risk events, agents are motivated to trade by stochastic incentives

[11]Any price is a result of a bilateral trade agreed by two agents in the markets.

with a mean shift following a Gaussian distribution. Agents trade widgets with others in a standard market microstructure mechanism, the continuous open-book system, which is the protocol utilized by most exchanges in the global financial market. The other variability in this paradigm sits on trading agents' *intelligence* level, reflected by their trading strategies and access to system-wide information. The concept of intelligence in agent-based modeling and its connection to the broader definition of intelligence in other fields will be briefly discussed in Chapter 2.

1.1 THESIS STATEMENT

With a brief conceptual background in play, I hereby highlight the central thesis of this dissertation.

Thesis Statement

This dissertation encapsulates a cascade of studies where the reward distributions of tasks are not limited to uniform or Gaussian distribution. At a high level, the task consists of an *environment* that emulates an actual financial market from the perspective of tail risk and *agents*, including AI and humans, who interact with the environment and among each other. In the first two studies, the narrative leans more on *agents vs. markets*. That is, we take an environment with tail risk as given and study the performance of individual AI and human agents against heavy-tailed rewards, with the focus on estimation efficiency. We focus not only on agents' ability to obtain optimal actions but also on whether their learning process towards convergence is efficient. In the third study, the environment does *not* entail tail risk by default. Instead, we introduce Gaussian incentives to agents who trade with each other in a market setting. We study whether leptokurtosis emerges as a result of agents' behavior and interaction.

1.2 THESIS OUTLINE AND CONTRIBUTION

The rest of the dissertation is organized as follows:

- Chapter 2 provides a brief introduction of artificial intelligence (AI) and machine learning (ML) for economics and finance. In this chapter, I formally define concepts like AI, ML, supervised, unsupervised learning, and deep learning. To start, I link two fields by crystallizing the definition of terminologies in order to disambiguate some confusions. I close this chapter by defining *intelligence* & *learning*, and their presence in economics & finance.
- Chapter 3 dives into the world of reinforcement learning (RL). The first part of this chapter serves as a crush course of (value-based) RL, deep RL, and distributional RL; it provides a detailed theoretical background to understand a critical framework that underpins AI and animal behavioral studies. In the second part, I discuss the limitation of the existing distributional RL algorithms using examples and known theories from classical statistics. Throughout this chapter, I show that economics & finance literature and AI literature can be welded into a unified framework.
- In Chapter 4, I show that both classical RL and distributional RL algorithms fail when rewards (returns) are affected by tail risk, i.e., leptokurtosis. I extend the distributional RL and expected reward RL framework, and introduce efficient estimation, while properly adjusting for differential impact of outliers on the two terms of the RL prediction error in the updating equations. I show that the resulting algorithm learns much faster and is robust once it settles on a policy.
- Chapter 5 presents a stochastic bandit experiment where the rewards of the arms are drawn from Gaussian, Student-t and Exponential distributions. In this chapter, I probe for the postulate that humans learn expected rewards by averaging experienced rewards and study human efficient value estimation in non-Gaussian environments. In addition to classical approach where human subjects report their choices, I customize the experiment such that human participants are required to periodically report their estimates. While I find substantial heterogeneity among participants, overall human reward learning appears to be guided by a desire to be efficient. This translates into enhanced choice confidence, except when participants show confusion as to the true form of the most efficient estimator.

- In Chapter 6, I propose a paradigm comprising a single-widget economy and a continuous open-book market to study economic welfare and the origin of tail risks. In this paradigm, the economy is populated with zero-intelligence agents, defined as rational agents who react to private incentives without utilizing economy-wide information such as order flow and trade prices. I then introduce a liquidity provider, who seeks to profit from learning economy-wide information, namely the recent evolution of bid-ask spread while attempting to remain inventory neutral. I demonstrate that in the absence of a liquidity provider, trading generates leptokurtosis and hence excessive tail risk. Introducing a profit-seeking liquidity provider further increases leptokurtosis, but the tail risk is not worsened.
- Chapter 7 concludes with reflections and future research directions.

Part II

FOUNDATIONS

2

ARTIFICIAL INTELLIGENCE AND MACHINE LEARNING FOR ECONOMICS AND FINANCE

[1]In this chapter, I wish to provide a condensed, non-technical introduction to a burgeoning topic: artificial intelligence (AI) and machine learning (ML) for economics and finance. The aim of this chapter is to (1) establish a foundation to link field of computer science and economics & finance, and (2) raise awareness of the *environment-agent* framework and the notion of *intelligent agents*. I start by defining terminologies from the field of computer science and then briefly discuss shared traits between ML studies and traditional economics & finance research. Finally, I shall introduce the concept of *intelligence* and *learning*.

2.1 CONCEPTUAL FOUNDATION

Artificial Intelligence (AI) studies the design of a system (computer) that can learn and act rationally (or like a human) through learning (McCarthy, 2007; Sutton, 2020)[2]. *Machine Learning* (ML) is a branch of AI on how machines can learn from data (Bishop, 2006; Goodfellow, Bengio, & Courville, 2016). In ML research, broadly there are three important paradigms: *supervised learning*, *unsupervised learning*,

[1]Part of this chapter was inspired by discussions with Professor Tibor Neugebauer during his visit to Melbourne. The author takes full responsibility of all views presented here.

[2]It is worth mentioning that this view comes from a computational modeling perspective. As I shall elaborate the discussion further in Section 2.2 the definition of *intelligence*, readers should be aware that even within the field of AI there has been an ongoing debate over the definition of AI. Readers who are interested in this topic may consult definition from other venues, e.g., S. Russell and Norvig (2020) and Nilsson (2009).

and *reinforcement learning* (Nilsson, 2009; Sutton & Barto, 2018)[3]. All three were inspired, and have inspired, computational neuroscience, since the three are important to understand neural processes associated with learning in animals, including humans (Ludvig, Bellemare, & Pearson, 2011; Poggio & Serre, 2013; Cichy & Kaiser, 2019).

In *supervised learning*, agents[4] are given a dataset where each set of features x is associated with a label or target y. These targets serve a supervision role to guide the agents. The goal is to learn the mapping from the features to its associated target:

$$y = f(x) \quad or \quad p(y|x) \tag{2.1}$$

A significant portion of research in economics and finance falls into this category (e.g. factor modeling) and function $f(x)$ is often assumed to be linear. There, the common research agenda is to establish an interpretable relationship between the feature set $\{x\}$ and the labels y.

In *unsupervised learning*, agents are given a dataset containing only features but no labels. The goal is to find structure or patterns in the dataset in the absence of explicit "label" supervision. For instance, we may ask the agents to learn a probability distribution that generates the dataset:

$$p(x) \tag{2.2}$$

Some popular techniques in financial econometrics, such as principal component analysis (PCA), fall into this category. In a nutshell, unsupervised learning attempts to distill as much information (or preserve maximum variance) as possible from the unlabeled data while reducing the dimension of the data.

[3]This categorization is contestable; audiences from computer science or control theory may object to this categorization. However, I would refrain myself from the debate on this controversial topic. Instead, I would simply point out that while reinforcement learning is indeed conceptually very different from the other two, this categorization is not unjustified: (1) the quote from Chapter 29.6 of Nilsson (2009): *"There is another style of learning that lies somewhat in between the supervised and unsupervised varieties ... this style of learning is called reinforcement learning or (sometimes) trial-and-error learning."* (2) the quote from Chapter 1.1 of Sutton and Barto (2018): *"We therefore consider reinforcement learning to be a third machine learning paradigm, alongside supervised learning and unsupervised learning and perhaps other paradigms."*

[4]An agent can be any decision-maker, including both artificial agents and human agents.

Deep learning describes a set of algorithms in ML. The structure of typical deep learning algorithms involves a stack of neural network layers, and each layer contains several numbers of neurons[5]. The scale of a deep learning algorithm can be incredibly wide, ranging from a simple multi-layer perceptron model to a mega model like the GPT-3 that entails 175 billion parameters. Deep learning is often confused with supervised/unsupervised learning, but they are orthogonal; one may utilize deep learning in the context of both.

Having the definition of terminologies, I can now briefly discuss the connection between economics & finance, and ML. One of the prominent goals in academic economics and finance research is to find a best-fit "model" and its parameters that explain the economy and the financial market. It is also often assumed that a natural derivative of the better "model" is the ability to predict the future[6]. From the perspective of environment modeling and prediction, the field of ML indeed shares a similar objective with the field of economics and finance: using historical data, one of the primary goals of ML research is to find an optimal model that makes robust out-of-sample predictions. The general perception is that more advanced non-linear ML models are contestable alternatives to traditional models in predicting financial market events (such as price volatility). However, lack of model interpretability (so-called a "blackbox" approach) has caused the use of ML models to be controversial (De Prado, 2018).

In addition to model interpretability issue, there is also one critical aspect of AI where recent economics and finance research appears to have missed: *agents' actions*. Neither supervised learning nor unsupervised learning alone could constitute a complete AI system. A revisit of the definition of AI reveals that there are two fundamental components in the definition of AI: *learning* and *action*. *Intelligence* requires learning autonomous actions. Without the ability to select and perform an action, an agent is merely a prediction machine (an oracle).

[5]See Google Tensorflow for an interactive example: https://playground.tensorflow.org/.
[6]Though, it is well known that a good backtest result does not guarantee robust predictions.

2.2 INTELLIGENCE AND LEARNING

It is of paramount importance to define the term *intelligence* before attempting to model an artificial version. Surprisingly, despite recent advances in AI, a satisfactory definition of *intelligence* is yet to be agreed upon among researchers (P. Wang, 2019). S. J. Russell (2010) surveyed the literature and argued that AI should think or act like humans or rationally. This view of human-like AI is debatable (see chapter 35 of Nilsson (2009)), and frankly in my humble opinion it is slightly abstract. McCarthy (2007) and Sutton (2020) provided an attempt from the perspective of a computation system:

> *"Intelligence is the computational part of the ability to achieve goals. A goal-achieving system is one that is more usefully understood in terms of outcomes than in terms of mechanisms."*

The above reward-centric definition clearly verges on *artificial intelligence* (Silver et al., 2021). I personally prefer the definition from the *Cambridge English Dictionary* (2022) since it encompasses the essence of *general intelligence* (see Figure 2.1): reasoning the environment and learning autonomous actions.

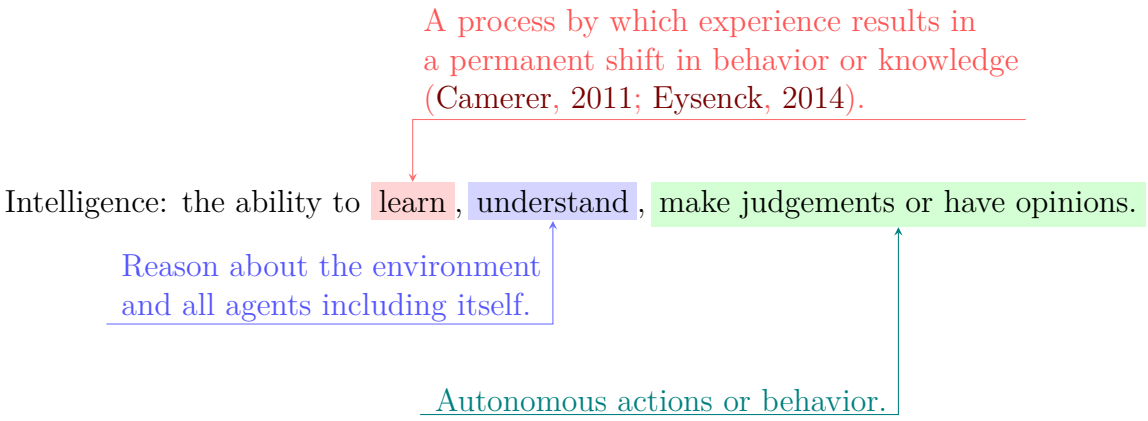


Figure 2.1. Definition of intelligence. Colored annotations are added by the author of this thesis. Source: *Cambridge English Dictionary* (2022).

An enormous amount of work in economics & finance has been devoted to understanding and reasoning about the environment[7]. One could argue that all supervised/unsupervised learning algorithms are essentially models for reasoning about the environment[8]; there, learning is reflected by information acquiring. As aforementioned, there is an obviously missing component that merits some attention: *learning actions*.

Formally, learning is a process by which experience results in a permanent shift in behavior or knowledge (Camerer, 2011; Eysenck, 2014). Experience can be generated through interaction with the environment: agents observe information, make decisions, perform actions and receive feedback from the environment; then, agents learn to continue to perform rewarding actions while avoiding non-rewarding or negative-rewarding actions. This iterative process, i.e., reasoning and learning from interacting with the environment, sits at the core of intelligence.

One of the most successful frameworks that models intelligence is reinforcement learning[9]. The following quote from Professor Yann LeCun illustrates the importance of RL in building a general-purpose AI:

> *"If intelligence is a cake, the bulk of the cake is unsupervised learning, the icing on the cake is supervised learning, and the cherry on the cake is reinforcement learning."*

The RL framework will be formally introduced in Chapter 3.

The notion of *environment-agent* (see Figure 2.2) is not alien to economics and finance whereas the notion of *intelligence* is not crystal clear. In mainstream neoclassical economics models, "representative agents", who were (mostly) assumed to be perfectly rational[10], were introduced as auxiliary tools for solving a market equilibrium (Chakraborti, Toke, Patriarca, & Abergel, 2011). However, the attention

[7] *"Much work in finance is devoted to identifying characteristics of firms, such as measures of fundamentals and beliefs, that explain differences in asset prices and expected returns."* (Koijen, Richmond, & Yogo, 2020)

[8] And perhaps planning, a crucial concept in intelligence modeling, but it goes beyond the scope of this dissertation.

[9] Another popular framework is Bayesian learning.

[10] An agent is "rational" if it follows the greedy algorithm; that is, the agent always chooses an action that maximizes its performance measure, such as a utility function (economics) or sum of expected reward (RL) (S. J. Russell, 2010).

was never on agents' learning but on the outcome of an equilibrium. Much research in economics and finance focuses exclusively on finding optimal models to describe various aspects of the economy and financial markets; agents' behavior and actions are exogenously postulated by the designer and hence not autonomous. But how does an equilibrium arise? Do agents know the optimal actions ex-ante to perform at the beginning as if the environment has alreaxdy reached equilibrium? The process of learning, or the path toward equilibrium, is elliptical.

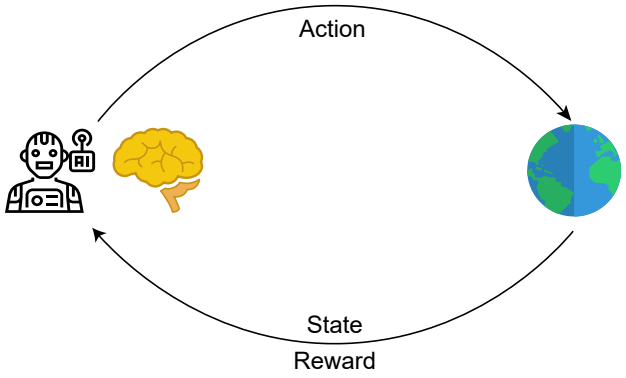


Figure 2.2. Agent-environment interaction[11]. Adapted from Sutton and Barto (2018).

Some game theorists have recognized the gap and probed for the missing component, i.e., agents' learning (Fudenberg, Drew, Levine, & Levine, 1998; D. Friedman, 1998; Camerer, 2011). Inspired partially by psychologists (e.g., Bush and Mosteller (1951) and Bush and Mosteller (1955)), different learning models were developed to unveil agents' learning process, reflected by acquiring new information, in a game theory setting[12]; one such model is reinforcement learning.

Most work of reinforcement learning in economics has paid attention to foundations and theories. Thus, the notion of reinforcement learning among game theorists and (mostly) behavioral economists preserves its primitive form: stimulus-response. Meanwhile, reinforcement learning outside the field of economics has drastically evolved (see Chapter 3 for more technical details), now combining reasoning and

[11]Source: Adobe image stock under Standard License; Robot, Brain, Globe.
[12]Evolution, belief updating (predictive coding), reinforcement learning, experience-weighted attraction, direction and rule-based updating. See chapter 6 of Camerer (2011) for details.

modeling ambient factors (deep learning) and dynamically learning policies (reinforcement learning). Arguably it is now framed as an overarching framework for modeling *intelligence* than just a stimulus-response model[13] for agents' learning.

Despite its successful application, reinforcement learning is only one framework to model intelligence and learning. Other research schemes on learning, for instance, *predictive coding and active inference*, have also been developed over the past several years (H. Brown, Friston, & Bestmann, 2011; Friston et al., 2016; Sajid, Ball, Parr, & Friston, 2021). There is no doubt that economists, animal behavioral studies, and AI researchers could benefit from each other by conducting cross-disciplinary research on learning. Indeed, learning theories drawn from game theory and behavioral economics have partially contributed to the success of recent AI like AlphaGo (Mnih et al., 2015; Silver et al., 2017). Meanwhile, perhaps economists may also cast their sights on recent progress in other disciplines when modeling their "agents", and regard learning as a critical component of studying the path toward equilibrium. I shall elaborate the discussion in Chapter 7 future directions.

The field of finance could also benefit from recent advances in RL and AI research, particularly when agents' learning and autonomous actions play vital roles in the research objectives (e.g., trading). One can view the financial markets as the environment and market participants as agents. Instead of imposing a strategy on an agent[14] *ex-ante*, we could ask the agent to learn/update its trading strategy on-the-fly as the market evolves. Better still, the RL framework serves as complementation rather than a complete revolution. Many problems in finance can be well integrated into the RL framework with little theory conflicts. For instance, concepts used to solve fundamental asset pricing models, such as the Bellman principle and dynamic programming, are the theoretical foundations underpinning reinforcement learning (e.g., see chapter 1 of Bossaerts (2005)).

Ramifications such as quantitative finance have long piqued interests in RL for its capability of (1) dynamically solving optimization problems with no analytical solutions (e.g., dynamic portfolio optimization), and (2) explicitly modeling an action space in the absence of labeled examples. Interests have grown exponentially

[13] In other words, many RL researches have gone beyond model-free learning.
[14] In RL terms, this is called agents' policy, see Chapter 3.

since deep RL thrives. This is mostly due to deep learning's ability to distill value or probability information from a large state space; such ability has enabled quantitative algorithms to process enormous amount of information for prediction while simultaneously updating their strategies in a consolidated framework. One blemish that has been quite heavily criticized by the traditional finance academics is the "black-box" approach of deep learning. Those who are lenient on deep learning have proposed methods to improve explainability of model and factor (see e.g., section 2.3 of Gu et al. (2020)). On the other hand, financial industry is fairly open to utilizing deep RL for control optimization; they perceive that deep RL is a double-edge sword. The success of AlphaGo has proven the ability of deep RL to find completely new strategies (policies) that human experts have never envisaged for thousands of years. The only concern from the industry practitioners and regulators is: how do we ensure that the new strategies do not incur catastrophic costs?

The span of RL application in quantitative finance is vastly broad, including portfolio optimization, option pricing, risk hedging, market making, high frequency trading, etc. For detailed reviews on algorithm specifics, I refer the readers to the latest reviews, Fischer (2018), Charpentier, Elie, and Remlinger (2021) and Dixon, Halperin, and Bilokon (2020). In my humble opinion, we should refrain from treating RL as merely a tool for optimization problems; understanding the crucial components of the RL framework and fully adopting the notion of environment-agent matter more than simply pruning the network structure of deep RL algorithms.

The notion of intelligence discussed so far is only at the individual level; intelligence may also appear at the group or system level. There is also a scheme of research in agent-based modeling called "zero intelligence (ZI)" that concerns intelligence emerging at the market level (Smith, 1962; Gode & Sunder, 1993). ZI agents are those who do NOT trade on system-wide information, such as trade price and volume but care only about local information. When a group of ZI agents trade in the same market, intelligence reveals at the system level in which stylized phenomenon of the real financial markets is reproduced. This notion of intelligence connects to *swarm intelligence* (Garnier, Gautrais, & Theraulaz, 2007; X.-S. Yang, Deb, Fong, He, & Zhao, 2016). For a detailed discussion on zero intelligence and swarm intelligence, I refer to readers to Chapter 6.

3

MODELING INTELLIGENCE: REINFORCEMENT LEARNING

This chapter provides a brief introduction of *reinforcement learning* (RL) and *distributional RL*. For a comprehensive review of RL, I refer the readers to Dayan and Niv (2008), Niv (2009), Glimcher (2011) and Sutton and Barto (2018); for a comprehensive review on distributional RL, I refer the readers to Bellemare, Dabney, and Rowland (2022). Towards the end of distributional RL, the depth of this introduction goes slightly beyond what is required to understand this thesis. Thus, while the materials are related at a conceptual level, readers who do not wish to understand the details of distributional RL are welcome to skim through the sections.

3.1 A GENTLE INTRODUCTION OF REINFORCEMENT LEARNING

Reinforcement learning is a computational paradigm that models agents' learning and decision-making through interactions with an environment (Sutton & Barto, 2018). Its theoretical foundation is Markov Decision Processes (MDP). An MDP is a stochastic control process that mathematically models the sequential learning and decision processes (Sutton & Barto, 2018). In this dissertation, I limit my discussions to discrete-time MDPs. See Definition 3.1 for a formal definition of a discrete-time MDP.

Definition 3.1: Markov Decision Process (MDP)

A tuple (S, A, R, P) characterizes a discrete-time MDP where:

- S : a set of states that describes the relevant information of the environment.
- A : a set of actions that describes all possible actions available to an agent.
- R : a reward function $S \times A \to \mathbb{R}$.
- P : a state-transition function $S \times A \times S \to \mathbb{R}$ that characterizes the transition probability $P(s^{'}|s, a)$ from a state $s \in S$ to another state $s^{'} \in S$.

The "Markovian" property of an MDP indicates that both the transition function P and the reward function R depend only on the current state of the environment and current action of the agent, but *not* the full historical trajectories. For time $t \in \mathbb{N}$,

$$P(s_{t+1}|s_t, a_t) = P(s_{t+1}|s_0, a_0, s_1, a_1, ..., s_t, a_t) \tag{3.1}$$

$$R(s_{t+1}|s_t, a_t) = R(s_{t+1}|s_0, a_0, s_1, a_1, ..., s_t, a_t) \tag{3.2}$$

While the assumption imposed in Equation 3.1 and Equation 3.2 may be perceived as rather stringently restrictive, the framework is in fact without loss of generality. A common practice to circumvent a non-markovian property, such as when the environment is auto-regressive in the state space, is to include the past $t \in \mathbb{N}$ steps into the state representation. Moreover, if necessary, one may also further assume that the state transition is independent of the agent's actions, $P(s^{'}|s, a) = P(s^{'}|s)$. For instance, in typical economics and finance research, individual agents (algorithm/investors) are often assumed to possess no significant influence on the market (Modigliani & Miller, 1958)[1].

A *reward functions* R characterizes positive or negative feedbacks received by the agents from the environment upon performing an action. It serves as the primary source to incentivize "learning". Iteratively, the agents should learn to perform

[1]Although from the GameStop short squeeze saga, it is obvious that collectively individual investors do have a significant influence on the financial market.

actions that offer positive rewards while avoiding the actions which offer negative rewards. Despite its significance in shaping agents' behavior, both the design of a reward function and its repercussion on agents' behavior have received limited attention. The reward function is perhaps one of the under-researched yet most controversial topics across RL, behavior and neuroscience fields (see Collins and Shenhav (2021); Silver et al. (2021)). In a typical RL system, a reward function is assumed to be either a win-or-lose style binary distribution (e.g. Silver et al. (2016)), or a probability distribution within the Gaussian space (e.g. Engel, Mannor, and Meir (2005)). A primary motivation of this dissertation is to relax this constraint and examine how agents, including RL agents and humans, react to non-Gaussian rewards.

Given an MDP, the goal of an RL agent is to learn what actions to take when observing a state to maximize the long-term expected reward. This notion of behavior is grounded in the algorithmic design of *policy function*, and the concept of "learning" is interpreted mathematically to updating a policy function by interacting with an MDP (see Definition 3.2).

Definition 3.2: Policy

Given an MDP, a policy function π is a mapping from states to probabilities of choosing an action, $a = \pi(s)$, $s \in S$, $a \in A$.

Definition 3.2 is known in economics as a *mixed-strategy* (e.g., see chapter 6 of Tadelis (2013)). Most economists would claim that, when you're playing against nature, pure strategies are optimal generically. However, the view ignores the process of "learning": you need mixing because exploration is crucial as part of learning. Exploration is there to learn, not only the optimal action given a state, but also to visit all state/action pairs so that the agent has a better sense of the environment itself (states, available actions, etc.). Without the latter, state transition P is never learned.

The design of a policy mapping leads to two main ramifications of RL algorithms: *policy-based* and *value-based* RL. A policy-based method attempts to directly model and optimize the probability distribution of each action $a \in A$ given a state in

$s \in S$ a parametric form with a set of parameters θ, $\pi_\theta(a|s), s \in S$. "Learning", or the evaluation and improvement of a policy, is typically achieved via gradient updates on parameters. In a value-based method, given a state s, agents choose actions based on its *value*, denoted as $Q^\pi(s,a)$ and calculated as the discounted sum of expected future rewards if the agent follows the same policy π onward (see Definition 3.3). "Learning" is achieved via computation or update of the values. The two approaches share common traits and face similar challenges. This dissertation focuses on value-based method.

Definition 3.3: Value Function

Given an MDP, an action-value function, $Q^\pi : S \times A \to \mathbb{R}$, is the discounted sum of expected future rewards if the agent takes an action and follows the same policy π onward.

$$Q^\pi(s,a) := \mathbb{E}_{\pi,P}[\sum_{k=0}^{\infty} \gamma^k R(s_{t+k+1}, a_{t+k+1})|s_t = s, a_t = a] \quad s_t \in S, a_t \in A \tag{3.3}$$

Similary, the value of a state, $V^\pi : S \to \mathbb{R}$, is defined as follows:

$$V^\pi(s) := \mathbb{E}_{\pi,P}[\sum_{k=0}^{\infty} \gamma^k R(s_{t+k+1})|s_t = s] \quad s_t \in S \tag{3.4}$$

where $\gamma \in [0,1)$ is the discount factor.

Under the value-based approach, the goal of an agent is to learn the optimal policy π where its action value $Q^\pi(s,a)$ is maximized:

$$Q^*(s,a) = \max_\pi Q^\pi(s,a) \tag{3.5}$$

and the optimal value of a state is the maximum of all action values in that state:

$$V^\pi(s) = \max_a Q^*(s,a) \tag{3.6}$$

3.2 TABULAR METHOD: Q-LEARNING

Mathematically, assuming that a time-invariant Markovian optimal policy exists, the goal of an RL algorithm is to find a solution to the optimization problem (Equation 3.5). In plain language, the goal is to find the best action given a state. In general, one should not expect an RL problem to be solvable without imposing many assumptions on the system, e.g. on state-action spaces (Nesterov, 1998). Even if an RL problem is solvable, hardly any analytical solution exists, and hence iterative methods are typically required.

One can simulate (e.g., monte-carlo) values for all state-action pairs to obtain the true environment dynamics (i.e. the state transition function P and the reward function R), yet generally speaking this is not feasible when state and action spaces are large; this method requires a task to be simulated to the end for infinitely many times[2]. Instead, "estimations" are much more feasible than simulations. In the "estimation" method, an agent estimates the environment dynamics based on its experience and assumes the estimation can be applied in the future. In this way, learning happens in the run-time: the agent is able to learn and perform actions sequentially. Simulation and estimation methods are practically not mutually exclusive; many RL algorithms utilize both methods (e.g., Alpha-Go uses monte-carlo tree search.)

The question of what functions are estimated sits at the core of RL algorithm design and contributes to two main classes of value-based RL algorithms, *model-free* and *model-based* methods (see Figure 3.1 for a taxonomy of different algorithms). In the *model-free* approach, either the policy π (*policy-based*) or the action-value function Q^π (*value-based*) is estimated directly. The agents do not care about the true underlying environment dynamics, such as the transition function P or the reward function R. Instead, the agent updates the action values directly by drawing from the experience that is collected through interacting with the environment. In contrast, in the *model-based* approach, the agents posit action values by explicitly maintaining estimates of the state transition function P and the reward function

[2]Note here there is another implicit assumption that the environment dynamics are stationary over time. However, the validity of this assumption depends on the nature of the task.

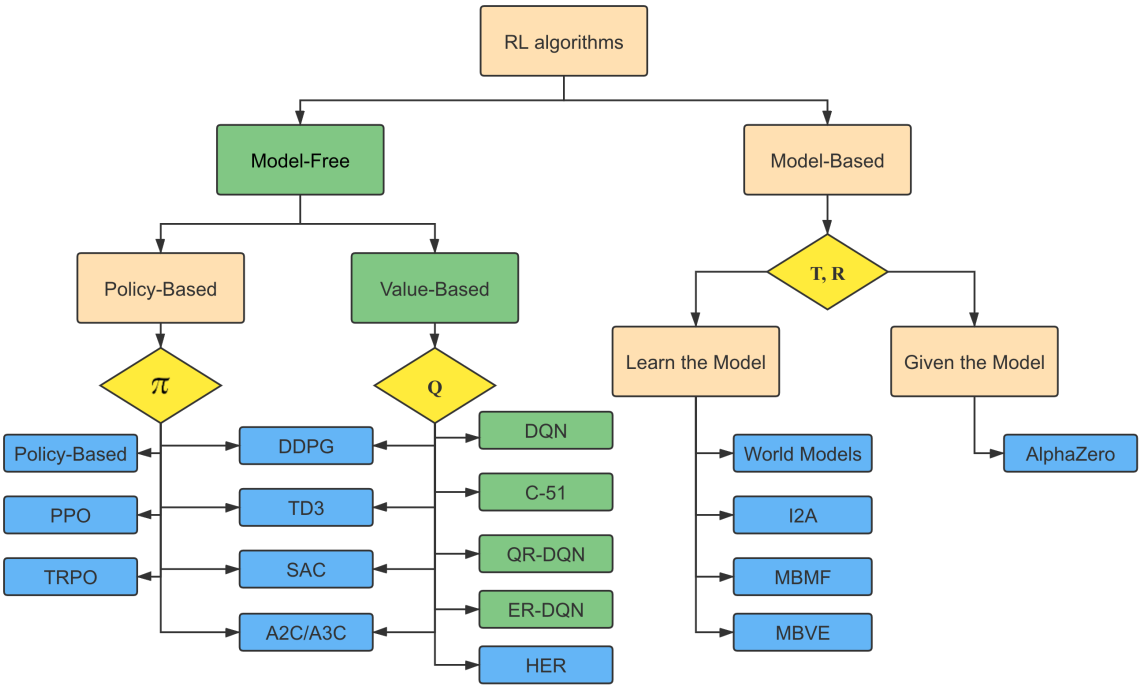


Figure 3.1. Taxonomy of reinforcement learning algorithms, sourced from OpenAI introduction of deep RL under the MIT License. Boxes highlighted in green (■) are the focus of this dissertation. Boxes highlighted in yellow (■) indicate the core functions to be estimated: policy function π, value function Q, state transition P and reward function R. Notice that it is practically infeasible to categorize all existing algorithms in a single diagram; the purpose of this diagram is to give a broad idea of the relationship between model-based and model-free RL algorithms, and between policy-based and value-based RL algorithms. Some RL algorithms do not have a clear-cut as to which category they belong to. In practice, many RL algorithms utilize the advantages of different approaches.

R (Sutton & Barto, 2018). These two functions are updated by sampling from the experience. The relationship between the two methods, as well as their roles in animal learning, have been one of the most discussed topics in computer science, behavioral and neuroscience studies. For discussions of the relationship and interplay between model-based and model-free methods, see the dissertation by Asadi (2015), and the survey by Moerland, Broekens, and Jonker (2020); for discussions from the behavioral and neuroscience perspective, see Niv (2009); Dayan and Berridge (2014); Daw (2012); Drummond and Niv (2020).

Model-free Q-Learning: Let us first consider a type of RL in which the agents do not explicitly model the environment dynamics. To elicit the details of "model-free" RL, here I introduce perhaps the most canonical RL algorithm, "Q-learning", by Watkins and Dayan (1992). Q-learning maintains an estimate of the Q-value function for each state-action pair and adaptively updates the Q values via *prediction errors* from the previous trial. Suppose we have a simple RL problem with two states and two actions, illustrated in Table 3.1. The goal of the agent is to find which action to take, either a_0 or a_1, under each state s_0 or s_1; or equivalently, which action value Q is higher among the two in each state. To begin with, the Q-learning algorithm initializes a Q-function based on some pre-specified protocol, e.g. initializes all Q-values with 0. Each time the agent encounters a state s, it will choose the action a based on a policy, such as the ε-greedy policy (Definition 3.4).

$Q(s,a)$	s_0	s_1
a_0	$Q(s_0,a_0)$	$Q(s_1,a_0)$
a_1	$Q(s_0,a_1)$	$Q(s_1,a_1)$

Table 3.1. An example of two-states two-actions RL problem.

Definition 3.4: ε-greedy Policy

For a probability $\varepsilon \in [0,1]$,

$$\pi(a|s) = \begin{cases} a = \arg\max_a(Q(s,a)), & \varepsilon \\ U\{a_i | a_i \in A\}, & 1-\varepsilon \end{cases}$$

At a time $t \in \mathbb{N}$, an action a_t is taken, the agent receives a reward $R(s_t, a_t)$ and a new state s_{t+1}. Together, the original state s_t, the new state s_{t+1}, the action a_t and the reward r_t constitute an *experience vector* (s_t, a_t, r_t, s_{t+1}).

Definition 3.5: Experience

An experience is a vector that contains s_t, a_t, r_t, s_{t+1}. An experience pool is a collection of experience vectors.

"Learning" is achieved via action-value updates based on the agent's most recent experience:

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[R(s_t, a_t) + \gamma \max_{a_i \in A} Q(s_{t+1}, a_i) - Q(s_t, a_t)] \tag{3.7}$$

where $\alpha \in [0, 1]$ is the learning rate (or the step size), $\gamma \in [0, 1)$ is the discount factor.

Here we assume that the one-step ahead update $r(s_t, a_t) + \gamma \max_{a \in A} Q(s_{t+1}, a)$ is a better estimate of the action-value $Q(s_t, a_t)$, i.e. we want the algorithm to update the action-value $Q(s_t, a_t)$ towards it. Hence, this term is referred to as the "target" action-value. The distance between the target action-value and the approximated action-value, $r(s_t, a_t) + \gamma \max_{a \in A} Q(s_{t+1}, a) - Q(s_t, a_t)$, is often referred to as the *prediction error* or the *temporal difference error*. The full pseudocode of Q-learning is presented in Algorithm 3.1.

Algorithm 3.1: Q-Learning

Input: α, γ, s_0, Q_0, ε

while *True* **do**
  $a_t \leftarrow \varepsilon$-greedy$(Q(s_t, a_t))$;
  $r(s_t, a_t)$, $s_{t+1} \leftarrow$ environment(a_t);
  $Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha(r(s_t, a_t) + \gamma \max_{a \in A} Q(s_t, a) - Q(s_t, a_t))$
end

Assuming an optimal policy π^* exists, one can prove that asymptotically Algorithm 3.1 converges to the true optimal action-value function Q^* (Watkins & Dayan, 1992) when the temporal difference error converges to zero. That is, the agent finds

optimal action-values. However, optimal convergence requires an important condition, among others, to be satisfied: all state-action pairs are visited infinitely often in the limit. Consequently, even though the choice of a policy function π is not fixed, exploration is required to be part of the policy. Furthermore, this condition raises the *exploration-exploitation* dilemma. "Infinitely often" is a theoretical term, and practically the agent should stop "exploring" when all state-action pairs are visited enough times. In RL research, the question of when exploration should stop is quite arbitrary, often set up by the engineers (e.g. exponential decay ε). In behavioral research, how humans balance the trade-off between exploration and exploitation is a trending research topic on its own (see e.g. Song, Bnaya, and Ma (2019)).

Moreover, the "target" action-value estimation in the Q-learning is considered as an "off-policy" learning. In this case, action values Q update does not necessarily have to follow the policy π, and thus these algorithms are called *off-policy* algorithms. For instance, one agent may learn a behavior policy that chooses the second-best action while updating the action-values Q using the optimal action value. This type of learning procedure is in contrast to the *on-policy* learning. In an on-policy learning, the target action-value estimation follows the behavioral policy. For example, Equation 3.8 depicts an on-policy learning algorithm.

$$Q(s_t, a_t) \leftarrow Q(s_t, a_t) + \alpha[r(s_t, a_t) + \gamma Q^{\pi}(s_{t+1}, a_{t+1}) - Q(s_t, a_t)] \tag{3.8}$$

Equation 3.7 and Equation 3.8 look almost the same, yet there are subtle differences between the two. The differences between the two types are beyond the scope of this dissertation. For a detailed comparison, please see Sutton and Barto (2018) Chapter 10 and 11.

3.3 FUNCTION APPROXIMATION AND DEEP REINFORCEMENT LEARNING (DQN)

While a discrete tabular state representation such as Table 3.1 is easy to implement, it becomes less efficient as the state space S and action space A grow larger or become continuous (Botvinick, Wang, Dabney, Miller, & Kurth-Nelson, 2020). Formally, we

say that tabular RL algorithms suffer from the *curse of dimensionality* (Sutton & Barto, 2018).

Moreover, under the standard MDP formulation, the state space S is assumed to be fully observable to the agents, and is mainly defined by the engineers or the experimenters. However, the implicit assumption here requires the set of all sensory inputs Ω from the environment to be the same as the state information S used by agents for decision-making. That is, although agents may receive multiple sensory inputs (visual, audio, tactile, etc.), only the relevant information is typically modelled into an MDP. Likewise, the action space A is typically designed to tailor to a particular task. In both RL and behavioral research, there are numerous attempts to address the question of how agents craft state and action representations; many models focus on state-space dimension reduction. For a review of modeling the state-action representation in animal learning, see Collins and Shenhav (2021); for a review of state-action abstraction in RL research, see Abel (2019).

One important question remains even if we manage to trim down the state space: what is the relationship between the state space and the action values? Table 3.1 is a tabular mapping, yet as we have discussed above, it is not always feasible to establish a tabular relationship between all states and actions. A natural way to generalize this relationship is to parameterize the Q values, leading to the *function approximation* approach (R. Bellman & Dreyfus, 1959; Samuel, 1967, 2000):

$$\hat{Q}(s, a) = f_a(\phi(s)) \tag{3.9}$$

where $\phi(s)$ is a state representation function, and f_a is a function that takes in the state presentation value and outputs the action-value for each action $a \in A$. For example, if we assume that there is a linear relationship between the action values $a \in A$ and the state inputs $s \in S$, then f_a is a linear function:

$$\hat{Q}(s, a; W) = W^T \phi(s) \tag{3.10}$$

where W is a weight vector that has the same dimension as the state representation vector $\phi(s)$.

The objective of function approximation algorithms is to minimize the TD error, namely the difference between the true value function (called the target value function) and the approximated value function:

$$\min_W Q_{target}(s,a) - \hat{Q}(s,a;W) \tag{3.11}$$

or equivalently, to minimize the square of the TD error:

$$\min_W \frac{1}{2}[Q_{target}(s,a) - \hat{Q}(s,a;W)]^2 \tag{3.12}$$

Equation 3.12 is a convex minimization problem, we may apply *stochastic gradient descent* to update the weights:

$$W \leftarrow W + \alpha[Q_{target}(s,a) - \hat{Q}(s,a;W)]\nabla\hat{Q}(s_t,a_t;W) \tag{3.13}$$

Since the target value function $Q(s,a)$ is generally unknown, practically it also has to be approximated. Similar to the canonical Q-learning, an approximation can be the sum of the current reward and the best action-value of the next state (Barnard, 1993; Sutton & Barto, 2018):

$$W_{t+1} = W_t + \alpha[\hat{Q}_{target}(s,a) - Q(s_t,a_t;W_t)]\nabla\hat{Q}(s_t,a_t;W_t) \tag{3.14}$$

where $Q_{target}(s,a)$ is approximated by $\hat{Q}_{target}(s,a) = r(s_t,a_t)+\gamma \max_{a\in A} Q(s_{t+1},a;W_t)$. At a first glance at Equation 3.14, one may become slightly confused with the "double approximations" since effectively the agent is evaluating its actions and learning its estimations of the action-values from its own "guess". This is mainly due to the lack of explicit "supervision" in an RL framework. Unlike in supervised learning where "correct labels" are available for the agents to approach, in the RL framework, there are only reward signals[3]. As a result, an RL agent has to find a *target* to learn from. As Sutton and Barto (2018) pointed out, an RL agent must "learn a guess from a guess". Iteratively, this "self-adversarial" nature, which sits at the

[3]Hence, some scholars consider RL as a semi-supervised learning method.

center of an RL algorithm, helps an RL agent to learn better policies from its own estimations.

In the function approximation method, "learning" is achieved via updates in the weight vector that corresponds to an action rather than directly via the action-value. By updating the weights that generate the action-value given a state, the agent is able to converge its current policy to a better one.

When the function f_a is a multi-layered artificial neural network, the RL system is known as deep reinforcement learning. Implementation of deep reinforcement learning varies, and some technical details are beyond this thesis. Without diving deeper into the technical details, here I will illustrate a famous example, Deep Q Network (DQN), in Figure 3.2 (Mnih et al., 2015).

3.4 DISTRIBUTIONAL REINFORCEMENT LEARNING

While a significant amount of effort has been spent on tweaking the neural network architecture to improve the algorithmic performance, virtually all value-based RL methods still rely on a core assumption (or convention): the action-value is directly estimated as a scalar quantity regardless of whether the algorithm is a tabular method or function approximation method[4]. Bellemare, Dabney, and Munos (2017) proposed that value-based RL algorithms should estimate a unique conditional probability distribution of the value for each state and action, instead of its mean value[5]. It is well known the entire distribution entails much richer information (higher-order moments) than its mean, and hence the distribution estimation is expected to yield better algorithmic performance. Table 3.2 illustrates an example distributional RL in the tabular form. $Z(s, a)$ is effectively the experienced action-value $Q(s, a)$ over

[4]This section benefits greatly from the discussions with Matthew Farrugia Roberts, as well as the class report produced by Matthew and his teammates (https://github.com/matomatical/coursework-portfolio/blob/main/farrugia2020expectiles-dopamine.pdf). The author takes full responsibility of all views presented here.

[5]To clarify, Morimura, Sugiyama, Kashima, Hachiya, and Tanaka (2010) and Morimura, Sugiyama, Kashima, Hachiya, and Tanaka (2012) were the first propose a distribution version of the value function, but Bellemare et al. (2017) were the first to apply the distribution concept in a control setting (i.e. with actions), see Bellemare et al. (2017) for more details.

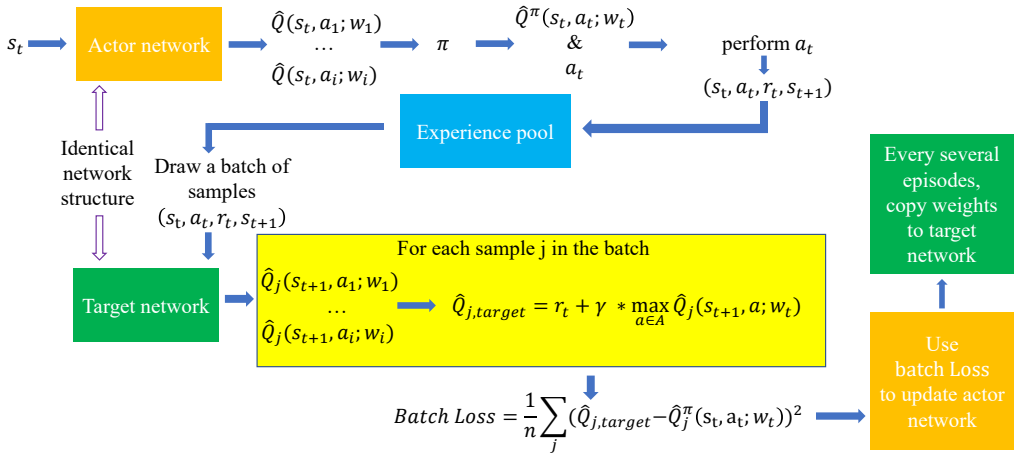


Figure 3.2. An example implementation of deep reinforcement learning. The actor network receives the state information and outputs a vector of action values for all actions. The agent selects an action according to the behavioural policy π. The agent collects an experience pool with a collection of experience vectors (s_t, a_t, r_t, s_{t+1}). The target network has an identical network structure (e.g. same number of neurons and network layers) with the exception of weight values. Whenever the experience pool is full, the agent draws a batch of samples from the experience pool and performs the temporal difference update. For each sample, the target network takes in the next state $_{t+1}$ and outputs the action values. Then an estimated $\hat{Q}_{target}$ is calculated based on the action-value, the corresponding reward and the discount rate (recall Equation 3.14). We can update the actor network weights using batch loss (e.g. stochastic gradient descent). See Mnih et al. (2015) for details.

time as one takes actions in different states. Here I emphasize that without imposing further assumptions, $Z(s, a)$ is inherently stochastic and non-stationary.

$Q(s,a)$	s_0	s_1
a_0	$Q(s_0, a_0)$	$Q(s_1, a_0)$
a_1	$Q(s_0, a_1)$	$Q(s_1, a_1)$

=>

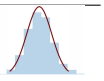
$Z(s,a)$	s_0	s_1
a_0		
a_1		

Table 3.2. An example tabular form distributional RL. Each action-value is represented by a distribution $Z(s, a)$ instead of a single quantity action value $Q(s, a)$. Here I used a histogram with an estimated probability density function (PDF) for illustration purpose.

Distributional RL algorithms outperform many RL algorithms in various complex and challenging games (Bellemare et al., 2017; Dabney, Ostrovski, Silver, & Munos, 2018; Dabney, Rowland, Bellemare, & Munos, 2018; Bellemare, Le Roux, Castro, & Moitra, 2019; Rowland et al., 2019). Moreover, the idea has spanned over to behavioral and neuroscience studies where researchers have found evidence that constructing action-values in a distribution rather than a single scalar quantity is biologically plausible (Dabney et al., 2020; Lowet, Zheng, Matias, Drugowitsch, & Uchida, 2020).

In the following, I will first briefly introduce each algorithm, and then highlight some key assumptions and issues in the empirical implementation.

3.4.1 *A broad scope of implementation*

In its vanilla form, the main structure of a distributional RL algorithm remains the same as the DQN implementation, i.e. two networks with identical network structure, TD updates, etc. The main difference is that the network outputs a conditional distribution of $Z(s, a)$ with which to estimate the state-action value $Q(s, a)$ rather than a point estimate.

The decision-making process does not drastically change per se because action values governs decision-making. The difference is in the *computation* of action value.

Each action-value is computed by integrating over its estimated distribution (i.e., the expected value of an action-value distribution). Hence, the main challenges are embedded within the distribution approximation. More precisely, the following two broad questions are to be considered:

1. How do we "approximate" an action-value distribution of $Z(s, a)$?
2. How do we embed the concept of action-value distribution of $Z(s, a)$ into the canonical RL framework?

The first question goes beyond the field of RL. In the *parametric* method, we assume that each distribution can be finitely represented by a known parametric function (e.g. Gaussian with mean and standard deviation, or student-t with degree-of-freedom, etc.). Finding a distribution is equivalent to finding the parameters associated with the engineer-specified distribution function. Some distributional RL studies belong to this branch and stick to a tight family of distribution functions, e.g. Gaussian distribution (Sato, Kimura, & Kobayashi, 2001; Engel et al., 2005; Ghavamzadeh & Engel, 2007; Morimura et al., 2012) or mixture of Gaussian distributions (Barth-Maron et al., 2018; Choi, Lee, & Oh, 2019). However, the success of parametric estimation depends heavily on the validity of distribution function assumption.

Alternatively in a *non-parametric* method, to approximate a distribution, we could start by generating statistical characteristics of a distribution via a given sample. Each distribution possesses certain statistics to characterize itself. In plain language, if we know some statistical properties of distribution, we know what this distribution roughly "looks like". For instance, a probability distribution function can be characterized by equal-sized bins with different heights (either in probabilities or frequencies), namely, a histogram[6]. The non-parametric method is more flexible yet computationally more difficult than the parametric method.

Notice the notion of parametric vs. non-parametric is limited to distribution only. Whether one chooses to use parametric methods (e.g. a neural network) to compute distribution function parameters or to use the statistics approach depends on the implementation, and is potentially relevant to the second question.

[6]Although in my opinion histograms are good for visualization, it is a suboptimal way of approximation, readers are encouraged to use quantiles whenever possible, see below for discussions.

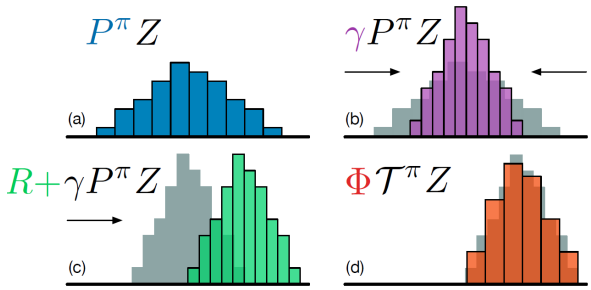


Figure 3.3. Categorical temporal difference update, reprinted from Bellemare et al. (2017) with permission under the Creative Commons Attribution 4.0 International License.

The second question is more involved in the context of reinforcement learning. Replacing the scalar action value with a distribution raises more questions in the implementation. For instance, how do we calculate the "target" action-value distribution like we do for a scalar action-value in Equation 3.7? Can we obtain convergence towards the true action-value distribution? What metrics should we use to calculate the distance between distributions? The answer to the above questions depends on algorithm specifications.

Although the actual implementation varies between algorithms, it is important to have these two top-layer questions in mind. Below, I will first lay out the details of three distributional RL algorithms from those two perspectives, then I will briefly discuss the limitations of distributional RL algorithms and their connections to classical statistics in Subsection 3.4.6.

3.4.2 *Categorical distributional reinforcement learning*

In so-called categorical RL (Bellemare et al., 2017), each action-value distribution is approximated by a histogram. Before I dive into the empirical details of this approach, readers are reminded that, from a pure statistical view, there is no compelling reason to use histograms as an estimation of the true probability density function (PDF). Histograms are **not** consistent estimates of the true PDF given a fixed sample size N, even if the true PDF is stationary. For a more detailed discus-

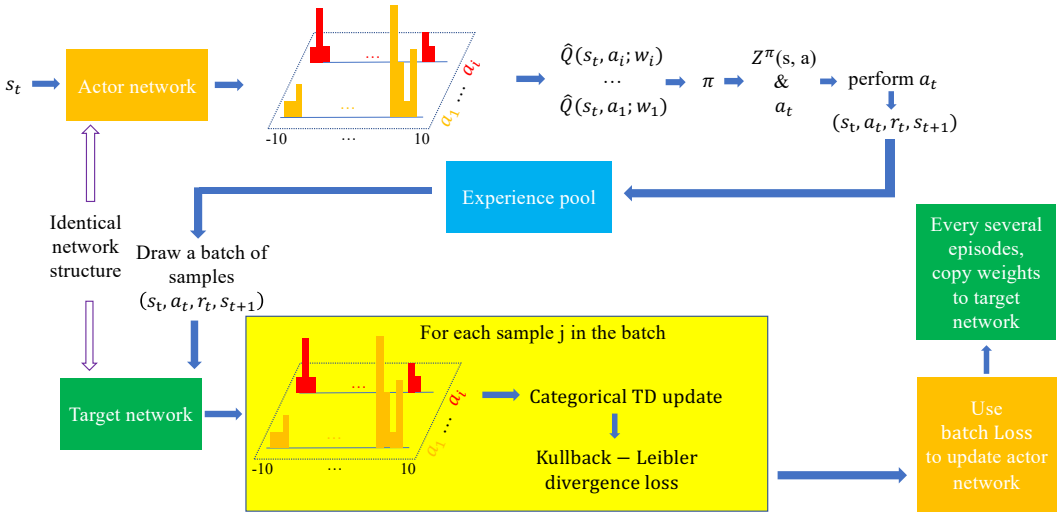


Figure 3.4. Categorical distributional reinforcement learning. Differ from DQN, the network outputs a matrix with the size of $N-1$ (number of bins) by $|A|$ (number of actions). Each action-value Q is calculated as the probability weighted average of its bins. The temporal difference update is illustrated in Figure 3.3, and the steps are laid out in Algorithm 3.2.

sion, I refer the readers to Chapter 14 (Estimating Distributions and Densities) of Shalizi (2022).

To characterize a histogram, we need two components: (1) bins and their locations on the x-axis, and (2) corresponding height (frequency/probability) for each bin on the y-axis[7]. In this case, we assume that the true action-value PDF is approximated by a truncated histogram between $[V_{min}, V_{max}]$, since the true PDF goes from $-\infty$ to $+\infty$ on the x-axis. Besides, we also assume that there are N equal-sized bins within the boundary. See Figure 3.5 for a mini example. In Bellemare et al. (2017), there are in total $N = 51$ bin edges (called atoms) on the x-axis (hence the name C51), and 50 bins in total. The vector output from the neural network contains the frequencies of each bin (i.e. the collection of height for each bin) for each action a_t, conditional on the state input s_t. Each action value $Q(s_t, a_t)$ is calculated as the probability-weighted average of each bin (i.e. the expected value or the mean).

[7]Imagine that you are drawing a histogram, these are the two necessary components.

To perform a temporal difference update, the algorithm first shrinks the histogram by γ and shifts the atoms (x-axis) by the reward amount r_t, then it re-distributes the frequencies back to where the original boundaries. The result distribution is the "target" action-value histogram. This "shrink, shift and re-distribution" process seems convoluted, yet it is required to address one of downsides of using a histogram; without re-distribution of the values, the histogram would immediately move out of the truncated range. The process is illustrated in the original paper (see Figure 3.3).

Loss for updating the weights of the neural network is calculated as the Kullback-Leibler divergence between the "actor" distribution and the "target" distribution. The choice was due to a technical issue in the implementation. In Bellemare et al. (2017), the authors used the Wasserstein distance to perform the policy convergence proof. However, the Wasserstein metric could not generally be minimized via stochastic gradient descent. Hence, Kullback-Leibler divergence was used in the empirical implementation. See Algorithm 3.2 for the pseudocode[8] and Figure 3.4 for an illustration[9].

The empirical limitation of this approach is obvious: how do we ensure that the true action-value lie inside the boundary? If not, then how do we determine the appropriate boundary in the first place? In practice, it is necessary to specify a wide enough range $[V_{min}, V_{max}]$ to partially mitigate this concern, but in the meantime one also needs a smaller bin size to be able to differentiate between distributions for different actions. In a nutshell, the balance between the maximum bin range and the bin size is critical in the empirical implementation.

[8]In line 16, m_l refers to the value from re-assigning the target distribution value according to index l. Note that although the term $u == l$ is omitted in their original pseudocode, in practice it is important to have a check whether u equals l and adjust the probability re-assignment accordingly. Otherwise, the sum of the resulting distribution m does not equal to one of the action value goes beyond $[V_{min}, V_{max}]$, leading to a distribution collapse/explosion.

[9]To visualize the categorical distributional RL in action, readers may visit https://www.youtube.com/watch?v=vIz5P6s80qA.

Algorithm 3.2: Categorical distributional reinforcement learning

Input: V_{min}, V_{max}, N

$\Delta B = (V_{max} - V_{min})/(N-1)$; // bin width

Initialize $atoms = [V_{min}, V_{min} + \Delta B, ..., V_{max} - \Delta B, V_{max}]$ where sizeof$(atoms) = N$;

while *True* **do**

  for *each action* a_i **do**

    $Z^{actor} = f_{a_i}^{actor}(s_t)$; // probability vector, size: N-1

    $Q(s_t, a_i) = \sum_j^n Z_j^{actor} * (atoms + \Delta B/2)$; // expected action-value

  end

  $a_t \leftarrow \varepsilon$-greedy$(Q(s_t, a))$;

  $r(s_t, a_t)$, $s_{t+1} \leftarrow$ environment(a_t);

  /* categorical temporal difference update */

  $atoms_{target} = [r_t + \gamma * atoms]_{V_{min}}^{V_{max}}$;

  $b = (atoms_{target} - V_{min})\Delta Z$;

  $m \leftarrow 0$; // size: N-1

  $Z^{target} = f_{a^*}^{target}(s_{t+1})$;

  for $k \in [0, ..., N-1]$ **do**

    $l = floor(b_k)$, $u = ceiling(b_k)$;

    $m_l \leftarrow m_l + Z_k^{target} * (u + int(u == l) - b_k)$;

    $m_u \leftarrow m_u + Z_k^{target} * (b_k - l)$;

  end

end

Output: $-\sum_i m_i \log p_i^{actor}$

3.4.3 *Quantile distributional reinforcement learning*

Inspired by statistics and econometrics, the notion of value distribution approximation has then been extended from the histogram approach to the quantile approach (Taylor, 1999; Dabney, Ostrovski, et al., 2018). In the Quantile distributional RL (called QR-DQN), each action-value distribution $Z(s, a)$ is approximated by a cumulative distribution function (CDF)[10]. The true *inverse CDF* can be characterized by the true quantile function or the entire (infinite) set of the quantile values. In general, the true quantile function is unknown, so it has to be approximated. In the case of distributional RL, we are interested in the conditional distribution of $Z(s, a)$ given state s and action a. Each set of discrete quantile values is one way to summarize and proxy this conditional distribution. Based on agents' experience,

[10]To visualize how the quantile method works in the Atari games, readers may visit https://www.youtube.com/watch?v=zdh_BTOcVYs.

quantile regression allows us to find the conditional quantile values (Koenker & Hallock, 2001). To allow for non-linear relations, a neural network, with the quantile regression loss function as its loss function, is used to generate the quantile values. Graphically, to characterize a CDF line via a set of discrete quantile values, we need two components: (1) the pre-specified quantiles on the y-axis, and (2) the corresponding value of each quantile on the x-axis[11]. See Definition 3.6 for the definition of quantile values, and Figure 3.5 for a mini example.

Definition 3.6: Quantile Values

Give a distribution μ and a asymmetric parameter $\tau \in [0, 1]$, the τth-quantile of a random variable X, $\epsilon_{X\sim\mu}(\tau)$, is defined as the minimizer of the quantile regression loss:

$$\epsilon_{X\sim\mu}(\tau) = \arg\min_{\epsilon} \mathbb{E}_{X\sim\mu}[[\tau \mathbb{1}_{X>\epsilon} + (1-\tau)\mathbb{1}_{X\leq\epsilon}]|X-\epsilon|] \tag{3.15}$$

where $\mathbb{1}$ is the binary indicator.

Equation 3.15 produces the value ϵ, so that weighted average of distances of observations X from ϵ is minimized; the weights are based on whether X is above ϵ or not. For example, if $\tau = 0.5$, one will get median: quantile regression finds the median by minimizing the average (absolute) deviation from ϵ; if $\tau = 0.9$, you should get the 0.9-th percentile. Here the equation minimizes the deviations from the 0.9th-percentile, but weigh observations above it 90% while observations below it get weight 10%.

In QR-DQN, the vector output from the neural network constitutes the quantile values for each action a_t, conditional on the state input s_t. Each action-value $Q(s_t, a_t)$ is the expectation computed from the estimated quantiles conditional on the state and action. If you view the quantile approach from the PDF's perspective, the quantile approach effectively constructs equal-height bins instead of equal-width bins where the bin locations are pre-specified by the engineers. In the quantile case, the density of a PDF is represented by how "clustered" the bin locations are.

[11]Similar to the PDF approach, imagine that you are drawing a CDF, these are the two necessary components to draw a series of coordinates to construct a CDF line.

The temporal difference update in the quantile approach is significantly less complicated than in the categorical approach. Since the distribution boundaries are no longer concerns with the quantiles, we may shrink (γ) the distribution and add a reward r_t directly on the quantile values. However, a strong assumption is required to perform the direct TD update on the quantile values. I will discuss the assumptions and limitations in Subsection 3.4.5.

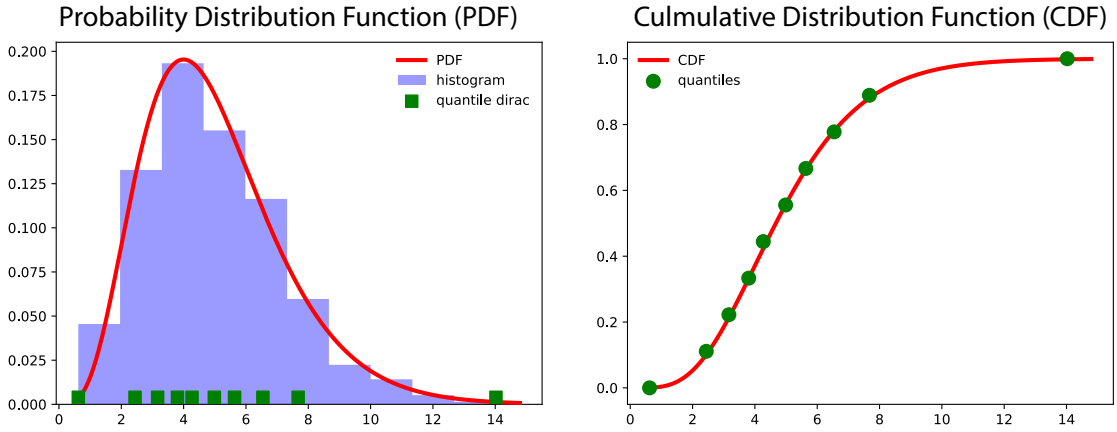


Figure 3.5. Histogram and Quantiles. The true underlying distribution is a gamma distribution with $\gamma = 5$. I randomly sampled 1000 values and plot the data. **Left:** the PDF is approximated by a histogram with 10 bins ($N = 11$ bin edges), truncated histogram boundary $V_{min} = 0.62$, $V_{max} = 14.03$, bin size $\Delta Z = 1.34$. Green square boxes show where the 10 equal-sized quantile values are located at. If you deem each box as an equal height "dirac distribution", the density is represented by how "clustered" these boxes are. **Right:** the CDF and its 10 equal-sized quantile values (equal-sized quantiles on the y-axis). Notice that these statistics are calculated from random samples for illustration purpose, whereas the task in distributional RL is to figure out the distribution (i.e. the samples) given statistics.

The QR-DQN algorithm requires pre-determined statistics τ, the number of points K and their positions. In principle, the more the number of statistics points we have, the better the approximations we obtain. However, this is not feasible in practice due to the modeling and computation capacity limitation. One extension to the existing quantile approaches is to leverage neural networks' model fitting ability

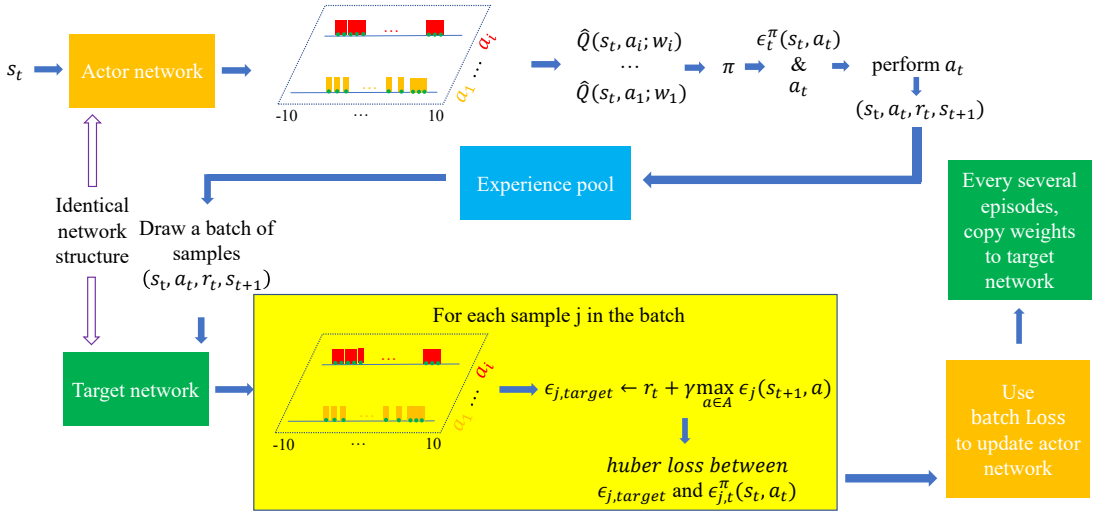


Figure 3.6. Quantile distributional reinforcement learning. Different from DQN, the network outputs a matrix with the size of number of quantiles by number of actions ($N \times |A|$). Here I emphasize the two major differences between this figure and Figure 3.4: (1) the temporal difference update is significantly more straightforward (with important assumptions highlighted in Subsection 3.4.5), and (2) the vector output of the network are the quantile values, i.e. bin locations on the x-axis. Each action-value Q is the mean of its quantile values.

to re-identify the statistics functions. This can be conducted either explicitly using another neural network (D. Yang et al., 2019), or implicitly by incorporating the functional approximation as part of the value learning network (Dabney, Ostrovski, et al., 2018).

3.4.4 *Expectile distributional reinforcement learning*

The expectile approach (called ER-DQN) is somewhat similar to the quantile approach. Expectile values are a family of summary statistics that generalizes the mean similar to the quantile values generalizing the median (Newey & Powell, 1987). While 0.5-quantile recovers the median, 0.5-expectile recovers the mean. See Definition 3.7 for the definition of expectile values. Appendix A4 and Figure 9 in Rowland et al. (2019) documented the difference between quantiles and expectiles.

Readers are reminded that there is no theoretical justification to expect that the expectile approach will perform better than the quantile approach in general. From statistics, we know that the efficiency depends on the underlying distribution (De Rossi & Harvey, 2009). In fact, if the mean of a distribution does not exist (e.g. Cauchy distribution), the expectiles may not even have a limiting distribution (e.g., a Cauchy distribution).

Definition 3.7: Expectile Values

Give a distribution μ and a asymmetric parameter $\tau \in [0, 1]$, the τth-expectile of a random variable X, $\epsilon_X(\tau)$, is defined as the minimizer of the expectile regression loss:

$$\epsilon_X(\tau) = \arg\min_{\epsilon} \mathbb{E}_{X\sim\mu}[|\tau \mathbb{1}_{X>\epsilon} + (1-\tau)\mathbb{1}_{X\leq\epsilon}|(X-\epsilon)^2] \tag{3.16}$$

where $\mathbb{1}$ is the binary indicator.

Prior to ER-DQN, the discussions on expectiles were mainly among the financial risk management literature (e.g. Taylor (2008); Bellini and Di Bernardino (2017); Daouia, Girard, and Stupfler (2018)) and the statistics literature (e.g. Kneib (2013); Waltrup, Sobotka, Kneib, and Kauermann (2015)).

Although expectiles are similar to quantiles in characterizing a distribution, the temporal difference update in the expectile approach cannot simply inherit the one in the quantile approach. The rationale behind this will be discussed in Subsection 3.4.5. In general, despite serving the same purpose of approximating a distribution, each type of distributional RL requires its own temporal difference update. Rowland et al. (2019) has provided a unified framework for different statistics. See Algorithm 3.3 for the steps.

Definition 3.8: Imputation Function

An imputation function is a function that takes in a set of statistics $\{\hat{\epsilon_1}, ..., \hat{\epsilon_K}\}$, and outputs a distribution with those statistic values (Rowland et al., 2019).

Algorithm 3.3: Unified temporal difference update (Bellman update)

1 *Imputation*: given the next state s_{t+1}, impute a set of estimated statistics $\{\hat{\epsilon_1}, ..., \hat{\epsilon_K}\}$ into a sample $[z_1, ..., z_N]$. Note that in theory N does not necessarily equal K, more discussions in Subsection 3.4.5;
2 *Temporal difference update*: calculate $z_i^{target} = r_t + \gamma z_i$, $i \in [1, N]$;
3 *Optimization*: compute a new set of statistics that can better characterize the samples $X = [z_1^{target}, ..., z_N^{target}]$, e.g. find X such that Definition 3.7 is minimized;

The authors emphasized that the temporal difference update should be performed on *samples* instead of its *statistics*. They consider the process of retrieving the samples from a set of statistics as the *imputation strategy*. Under certain assumptions, the temporal difference update may appear to be conductible directly on statistics, yet empirically they often result in a catastrophic collapse of the value distribution except some very special cases (Rowland et al., 2019).

3.4.5 *Discussion on samples and statistics*

While the discussion on statistics and the tabular RL examples in the literature were comprehensive and helpful, I believe that an example is necessary to crystallize the difference between *statistics* and *samples*.

Suppose that we have the following three samples μ_1, μ_2, and μ_3. Table 3.3 shows the key statistics of the three samples and Figure 3.7 illustrates the CDF.

$$\mu_1 = [1, 1, 2, 2, 4, 4, 6, 8, 8, 8, 8, 10, 10]$$
$$\mu_2 = [1, 2, 6, 8, 10]$$
$$\mu_3 = [1, 1, 1.5, 2, 6, 6, 6, 8, 8, 8, 9, 10, 10]$$

An important statistical concept is exemplified here: two samples are not necessarily identical despite having certain statistics to be exactly the same, whereas it is true conversely. This notion of non-unique samples raises problems in the afore-

	μ_1	μ_2	μ_3
N	13	5	13
median	6	6	6
mean	5.54	5.40	5.88
std. dev.	3.22	3.44	3.27
$\tau \in [0, 1]$	$[0.25, 0.50, 0.75]$	$[0.25, 0.50, 0.75]$	$[0.25, 0.50, 0.75]$
τ-th quantile	$[2, 6, 8]$	$[2, 6, 8]$	$[2, 6, 8]$
τ-th expectile	$[4.00, 5.54, 7.04]$	$[3.67, 5.40, 7.00]$	$[4.17, 5.88, 7.30]$

Table 3.3. Statistics of the three samples μ_1, μ_2 and μ_3.

mentioned *imputation strategy* (Rowland et al., 2019)[12]. If you calculate statistics from a sample with a fixed size N, given the locations of the statistics τ (e.g. 0.5th quantile), there exists a unique set of statistics to characterize that sample. However, if you impute a sample from a set of statistics with size K, theoretically speaking there could exist an infinite number of samples, even if you assume that all samples are of the same size (μ_1 and μ_3 have the same sample size). One possible way to ensure the uniqueness of the imputed sample is to impose a strong assumption that the number of samples $N \in \mathbb{N}$ equals the number of (equal-spaced) statistics $K \in \mathbb{N}$. This assumption is mentioned briefly in Dabney, Rowland, et al. (2018) and implied in Algorithm 2 of Rowland et al. (2019)[13].

Let us magnify the above example further. Notice that the sample μ_2 is equivalent to its $[0, 0.25, 0.5, 0.75, 1]$-th quantile values. Recall that in QR-DQN, we were able to perform temporal difference updates directly on the quantile values, whereas

[12] This twitter thread by Professor Timothy Gowers also nicely illustrates the concept of imputation strategy and its problem. The following math question was asked to his young daughter: a set of number has (1) mode of 24, (2) median of 21, (3) mean of 20, what are the 5 numbers? As he pointed out that the answer to this question is not unique.

[13] On a more practical matter, Rowland et al. (2019) mentioned that the imputation strategy could be conducted via either the minimization method (equation 8 of the paper) or the root-finding method (equation 7 of the paper). In the root-finding method, we seek to find a sample such that the Jacobian vector converges to $\vec{0}$. In Appendix D, the authors mentioned that they used the Scipy *root* function with the default parameters for empirical implementation. It is worth noting that by default, Scipy root employs "hybr" method, which requires the input shape (sample size N) and the output shape (shape of the Jacobian vector, which is equivalent to the number of expectiles K) to be the same. For more information on Scipy root function, see the documentation in `https://docs.scipy.org/doc/scipy/reference/generated/scipy.optimize.root.html`

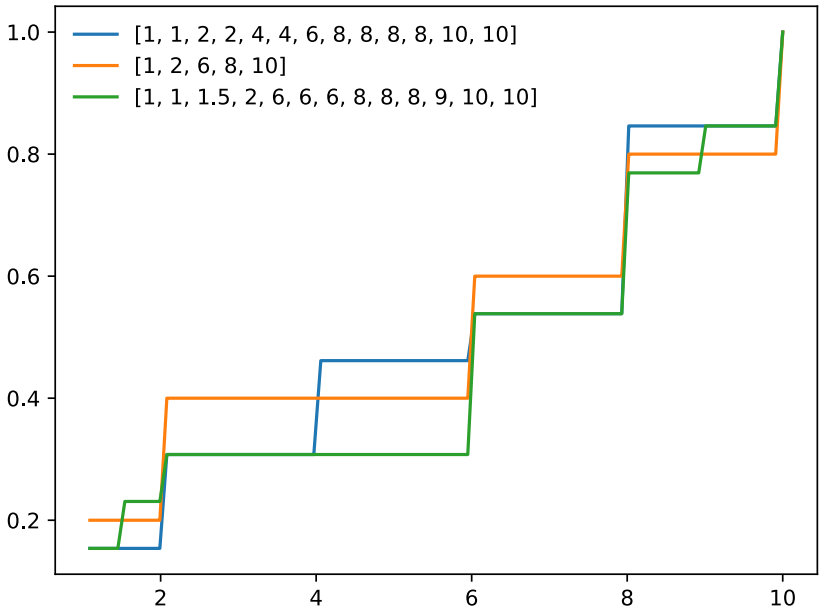


Figure 3.7. Empirical CDF of μ_1, μ_2, μ_3.

later we emphasized that the temporal difference updates should only be conducted on the samples (distributions) rather than on the statistics. This example illustrates that the quantile approach is indeed a special case: when the number of samples N is assumed to be equal to the number of (equal-spaced) quantiles K, the sample and the quantile values are identical.

3.4.6 *Discussion on limitations of distributional RL*

In summary, there are two layers of tasks in distributional RL: (i) decipher the true relation between the summary statistics and state-action pair (s, a), and (ii) identifying the true conditional action-value distribution $Z(s, a)$ using statistics. There are interesting topics in both tasks.

Task (i) is a function approximation task; it can be approached by a neural network. Distributional RL has its own unique problems. For instance, there is no guarantee that the output values from the neural network are monotonically increasing, which violates the definition of statistical values on the probability space. With quantile, this is known as the *crossing quantile curve* in the statistics literature (Kneib, 2013; Waltrup et al., 2015). Although this crossing quantile issue seemed to be under-appreciated by the distributional RL community at first, a recent paper by Zhou, Wang, and Feng (2020) proposed a solution to enforce the monotonicity of the output quantile values[14].

Issues in task (ii) have been addressed more in classical statistics. Broadly, within classical statistics, two ramifications are related to distributional RL: (1) statistical inference and (2) statistical learning theory.

Statistical inference concerns distribution identification, i.e. as more sample values are collected, can we identify its true underlying distribution (Casella & Berger, 2021)? The convergence from the sample to the true underlying distribution relies on the *Glivenko-Cantelli Theorem*: as sample size $N \to \infty$, the empirical CDF converges uniformly to the true CDF almost surely (as well as its statistics, such as quantile values). However, the results cannot be easily generalized to histograms; more assumptions are required to obtain convergence (e.g., kernel methods). Hence, intuitively it is more reasonable to use empirical CDF than a histogram to approximate an action-value distribution.

Statistical learning theory concerns estimating functional dependency based on a sample with a fixed size N (Vapnik, 2013). From a machine learning's perspective, statistical learning theory deals with supervised learning problems: given a sample with size N, we have a family of distributions (models) as candidates, and the question we ask is which distribution, out of the pre-specified family, has the highest probability to be the true underlying distribution? Model selection error, i.e., the probability of choosing a wrong model, is crucial. To minimize the model selection error, *Vapnik-Chernovenkis Theorem* is required, and the notion of *sample complexity* is also relevant.

[14]However, the solution is slightly brute-force.

To some extent, the problems in the distributional RL are closely related to the aforementioned ramifications with a slight different focus. On the one hand, the imputation strategy attempts to identify **an** action-value distribution from a set of statistics of size K, generated by a parametric neural network, rather than from a given sample. On the other hand, sample size N is fixed and pre-determined by the engineers. Notice that in this case, the uniqueness of the sample becomes a crucial problem; one cannot obtain a unique sample unless imposing heavy assumptions on the sample size N. If you view these assumptions from the perspective of statistical learning theory, the approaches restricted the family of distribution candidates for the action-value distributions (effectively down to one possible candidate if $N = K$). Yet there exists a dilemma between the choice of N and K: if K is too small, the quantile estimation becomes miserable when $N = K$; if we choose a large $N >> K$ while keeping K small, the Vapnik-Chernovenkis dimension of the space of possible true action-value distributions is large, with high probability, we will not be able to identify the correct (conditional) action-value distribution $Z(s, a)$; if we start with an arbitrarily large K, computationally it is so costly that it raises a question as to whether tracking the entire action-value distribution for each state-action pair is efficient and necessary.

In general, one cannot identify true action-value distribution $Z(s, a)$ using the above approaches when only summary statistics are available. The observations (e.g. quantiles) do not allow you to identify the true distribution with the sample size is limited and small (when $N > K$). This issue is at the heart of "sample complexity": you need far bigger samples (more quantiles) in order to identify the distribution.

One could argue that, unlike supervised learning, the ultimate goal of an RL system is to find the optimal policy through iterative Bellman updates, and hence whether it is even necessary to identify the true action-value distribution? Ideally, we want to identify the true action-value distribution to obtain a more non-fragile out-of-sample inference. Although it is true that one should not expect that this ideal case is generically possible, infeasibility should not prevent distributional RL algorithms from utilizing the existing tools in classical statistics to establish a more robust iterative policy update. This is one of the motivations of this dissertation.

Much work is needed for a more comprehensive theoretical analysis of the efficacy of the distributional RL algorithms. It is not obvious to prove that distributional RL is theoretically better than mean estimation RL universally, due to the difficulty in establishing a closed-form relationship between expected returns of an MDP in the context of a state-action value distribution (Rowland et al., 2019). In fact, some variants of distributional RL perform exactly the same both in tabular setting and in linear function approximation setting (Lyle, Bellemare, & Castro, 2019). This is not surprising since in an environment where there exists a sufficient statistic for the mean, and the statistic is efficient and can be updated recursively, it is not necessary to keep track of the entire distribution. In a Gaussian world where $Z(s, a)$ is distributed normally, canonical RL is computationally the easiest.

Part III

REINFORCEMENT LEARNING UNDER TAIL RISK

4

REINFORCEMENT LEARNING UNDER TAIL RISK

4.1 INTRODUCTION

Reinforcement Learning (RL) has been successfully applied in diverse domains. However, the domain of finance remains a challenge. A statistical feature central to finance is tail risk, or in technical jargon, *leptokurtosis*. For example, daily returns[1] on the S&P500 index, despite its broad diversification, follow a distribution with a kurtosis of 10 or higher, compared to the Gaussian kurtosis of 3. Kurtosis of individual stocks can be as high as 15 (Corhay & Rad, 1994).

In a leptokurtic environment, outliers, defined as observations in the tails of the distribution, are frequent and salient. This contrasts with infrequent, non-salient outliers, known as "black swans" (Taleb, 2007), or infrequent, salient outliers, known as "freak events" (Toenger et al., 2015). Statistically, the excessive mass in the tails of a leptokurtic distribution is compensated for by reducing mass about one standard deviation away from the mean (provided the standard deviation exists). As a result, besides outliers, small changes are also relatively more frequent than under the Gaussian distribution. Outliers are all the more salient since "typical" outcomes tend to be small.

RL is a key technique in machine learning. The goal is for an artificial agent to learn the right actions in a dynamic context. The agent is to search for the

[1]The term "return" refers to the percentage investment gain/loss obtained over a certain time frame (e.g. daily). It should not be confused with the term "return" referred to in the RL literature. However, the two terms are related as they both refer to some feedback from the environment/market to agents/investors. Likewise, we should be aware of the context (finance/RL) for proper interpretation of the terms "payoffs" or "rewards."

optimal actions as a function of the state of the environment. Effectively, the agent performs stochastic dynamic programming. In the most popular version of RL, Temporal Difference (TD) Learning, the agent starts with estimates of the values of all possible actions in all possible states, referred to as "Q values." The Q value of a state-action pair is the expected sum of future rewards conditional on taking suitable actions from the subsequent trial (trial 2) onwards.

In each trial, the agent tries an action, observes an immediate reward, as well as the state of the subsequent trial. Using adaptive expectations, the agent updates the estimate of the (Q) value of the action it just took in the state it was in. The new estimate is a weighted average of the old estimate and the "prediction error." The prediction error equals the difference between the old estimate, on the one hand, and the sum of the reward just obtained and the estimated value of the new state for a suitably chosen action, on the other hand. The weight on the prediction error is referred to as the "learning rate." In this paper, we focus on one version of TD Learning, called SARSA,[2] whereby the agent takes the action in the subsequent trial to be the one deemed optimal given the new state, i.e., the action that provides the maximum estimated Q value given the state.

A more recent version of RL learns Q values, not through adaptive expectations, but by remembering the entire distribution of the rewards in a trial and estimated Q values in the subsequent trial.[3] New estimates of the Q values of action-state pairs are then obtained by simply taking the expectation over this empirical distribution. This technique, referred to as *Distributional RL* (disRL), has been more successful than the traditional, recursive TD Learning, in contexts such as games where the state space is large and the relation states-action values is complex. See, e.g., Bellemare et al. (2017); Dabney, Ostrovski, et al. (2018); Rowland, Bellemare, Dabney, Munos, and Teh (2018); Lyle et al. (2019).

In a leptokurtic environment, TD Learning and certain versions of disRL are not robust. We show here that Q value estimates are very sensitive to outliers, and lead to frequent changes in estimated optimal policies, even after substantial learning.

[2]SARSA is short for State-Action-Reward-State-Action.

[3]The distribution can be remembered parsimoniously in various ways, e.g., as histograms, in terms of a specific set of quantiles or truncated expectations ("expectiles"). Compare, e.g., Bellemare et al. (2017); Dabney, Ostrovski, et al. (2018) and Rowland et al. (2019).

Consequently, if the learning rate decreases too fast, the agent's policy is unlikely to be optimal. If the learning rate is allowed to decrease only if the optimal policy remains unaltered, then the learning rate may never decrease since outliers continue to affect the estimated Q values, and hence policy.

We propose, and test, a solution. Exploiting the fact that disRL keeps track of the empirical distribution of estimated Q values for a given state-action pair,[4] we propose not to estimate the true Q value by simply averaging over the distribution. Instead, we propose to use an *efficient* estimator of the mean. Efficient estimators are those that minimize the standard error. When rewards are Gaussian, the sample mean is the most efficient estimator of the true mean. If one posits that rewards are generated by a t distribution with low degrees of freedom, one of the canonical leptokurtic distributions, a much better estimator exists. This estimator weighs observations depending on how much they are in the tails of the empirical distribution. The weighting is chosen to maximize statistical efficiency. The weighing does not simply truncate observations, but maps outliers back to the middle of the distribution.

When the true mean can be estimated using maximum likelihood estimation (MLE) and MLE provides consistent estimates, the MLE is the most efficient possible. Technically, it reaches the "Cramér-Rao lower bound." This is the case for the t distribution, so in our implementation, we use the MLE estimator. In general, the MLE estimator may not exist, and alternative estimators have to be found. We provide an example in the Conclusion.

The importance of using efficient estimators in finance, especially of mean returns, has been pointed out before. Madan (2017), for instance, shows how a kernel-based estimator generates lower standard errors than the usual sample average. Here, we propose to go for the *best* possible estimator, i.e., the one that maximizes efficiency (minimizes standard error).

Efficient estimation is not the only way disRL needs to be adjusted. Equally important is the following. The effect of leptokurtosis on experienced Q values decreases over time, as the agent re-visits trials in the same state and with the

[4]Technically, this is not exactly true for many versions of disRL in the literature. Only parametrically fitted histograms, quantiles or expectiles are remembered, reducing memory load.

same action. At the same time, leptokurtosis continues to impact the distribution of immediate rewards. This calls for decoupling of the two terms in the prediction error used in updating. We propose a simple way to implement the de-coupling. We show that it works effectively. We refer to our enhanced disRL as "efficient" disRL, and use the abbreviation *e-disRL*.

Using a simulation experiment, we prove the superiority of e-disRL over TD Learning and disRL. We also show superiority when rewards are drawn, not from a t distribution, but from the empirical distribution of daily returns on the S&P 500 index. We keep the environment in our experiment as simple as possible, in order to enhance transparency. We use a minimally complex environment, with two states and two possible actions. Optimal actions change with states. We envisage a situation where the artificial agent is asked to switch between two investments, while the optimal investments change with the state of the economy. Technically, our setting is a contextual two-arm bandit.

The framework may appear simple, but it is generic. The contextual two-arm bandit can readily be extended to handle more involved, and hence, more realistic situations, by augmenting the state vector or the number of states, and/or increasing the number of control options beyond two (arms). The bandit does not have to be stationary; it can change randomly over time, to form a so-called restless bandit. Continuous states and large state spaces can be accommodated through deep learning (Mnih et al., 2015; Silver et al., 2016; N. Brown & Sandholm, 2019). We chose a simple, canonical setting, in order to illustrate how easy it is for traditional RL to fail under leptokurtosis, and how powerful our version of distributional RL is to address the failure.

One could argue that there are other solutions to the problems that leptokurtosis causes. This could be GARCH or stochastic volatility modelling (Simonato, 2012), Monte Carlo approaches (Glasserman, 2013), moment methods (Jurczenko & Maillet, 2012), or parametric return process approximation and modelling (Nowak & Romaniuk, 2013; Scherer, Rachev, Kim, & Fabozzi, 2012). These procedures would effectively filter the data before application of RL. But it is known that mere filtering, while alleviating the impact of leptokurtosis, does not eliminate tail risk. Indeed, the filtered risk appears to be best modeled with a t distribution (which we

use here), or the stable Pareto distribution (for which variance does not even exist). These distributions still entail tail risk. See, e.g., Simonato (2012); Curto, Pinto, and Tavares (2009).

More importantly, none of the aforementioned procedures deals with control, which is what RL is made for. The procedures aim only at forecasting. As such, they do not provide a good comparison to RL. RL is engaged in forecasting as well, but *prediction subserves the goal of finding the best actions.* The problem we address here is whether leptokurtosis affects discovery and maintenance of the optimal policy, not merely that of finding the best prediction of the future reward.

Tail risk is a problem outside finance as well. In one very important context for RL, tail risk emerges when rewards occur only after potentially long chains of events (called "eligibility traces;" Sutton and Barto (2018)). The long chains cause the reward distribution to be leptokurtic. Singh and Dayan (1998) demonstrated that traditional TD Learning performs poorly when credit for rewards may have to be assigned to events that are at times too far in the past. Tail risk does not only plague learning by artificial agents. Evidence exists that humans, even professional traders, and despite continued vigilance, over-react to the frequent outliers that leptokurtosis entails (d'Acremont & Bossaerts, 2016).

For readers who may not be familiar with machine learning, we first provide a nontechnical description of the various machine learning techniques, honing in on reinforcement learning (RL), which is what this paper is about. We then explain intuitively how our enhancement of disRL generates robustness when tail risk affects the rewards. In Section 3, we introduce RL in a technical way. Section 4 then discusses the implications of leptokurtosis for TD Learning and disRL. Section 5 explains our solution. Section 6 presents the results from our simulation experiments. Section 7 concludes.

4.2 NONTECHNICAL OVERVIEW

4.2.1 *Machine Learning*

Broadly speaking, there exist three types of machine learning. All three were inspired, and have inspired, computational neuroscience, since the three are important to understand neural processes associated with learning in animals, including humans (Ludvig et al., 2011; Poggio & Serre, 2013; Cichy & Kaiser, 2019).

The first type is *supervised* learning, where the agent is given a dataset with cases (petal shape, color, etc.), described in terms of various features (e.g., flower features). Each case is labelled (e.g., "tropical flower," "temperate-climate flower"), and the goal is to learn the mapping from features to labels. The agent is given a limited set of cases with the correct labels (the "training set"), whence the term *supervised* learning. The mapping from features into labels can be highly nonlinear. A neural network with multiple ("deep") layers allows the agent not only to flexibly capture nonlinearity in the relationship between features and labels; it also provides a framework within which numerical optimization can be executed efficiently despite the nonlinearity and despite the high number of parameters.

The second type is *unsupervised* learning, where the agent is given a dataset containing features but no labels. The goal is to find structure, or patterns in the data. Techniques include factor analysis and cluster analysis. The retained factors or clusters help to identify hidden properties of the data, in the form of commonalities of features across data subsets. In other words, the agent is asked to come up with its own labelling system. It may end categorizing flowers into "tropical" and "temperate-climate" unless there are more relevant ways to cluster them.

4.2.2 *Reinforcement Learning*

We will be concerned here with a third type of machine learning, namely, *reinforcement learning*.[5] There, an artificial agent is effectively asked to do stochastic dynamic programming, i.e., to find, given the state of the environment, the actions

[5]It is also referred as semi-supervised learning as it sits in-between the above two types.

which maximize total rewards (or minimize total losses) for the foreseeable future. An example is when the agent is asked, every month, to decide between a stock index or Treasury bonds, as a function of the state of the economy, with the goal of maximizing lifetime gains. In the sequel, we will only look at maximizing rewards, since minimization of losses is isomorphic.

In stochastic dynamic programming, the agent recognizes that there are: *states*, maybe hidden; actions, which entail rewards and sometimes state transitions as well; and observables, including the rewards received. In its simplest form, the agent observes the states (observation may only reveal some properties of a state) and is asked to maximize the expected sum of (discounted) rewards in the future. Risk aversion can be built in by transforming rewards with a strictly concave function, as is standard in expected utility theory (Savage, 1972). Under risk aversion, the agent maximizes the expected sum of non-linearly transformed rewards. Here, we will assume risk neutrality, without loss of generality since we are focusing on learning. We will also assume that the states are fully observable. In a finance context, one could think of states as the general condition of the economy, measured through, e.g., change in industrial production.

Stochastic dynamic programming is immensely difficult even in the simplest cases. In theory, the agent has to find a *value function* (referred to as the Bellman function) that maps the state into the expected sum of rewards for optimal choices of actions. A key result in stochastic dynamic programming is that, under certain condition, the optimal policy is to pick the action that maximizes the rewards over the immediate trial plus the (discounted) expectation of the value function for the subsequent trial. (We will ignore discounting since it is not relevant to our argument.) Given the state a given trial is in, the value function can then be calculated by adding, to the immediate reward, the expectation of the value function at the next trial, provided the agent picks the action that maximizes both terms. This is referred to as the "Bellman equation." Across trials, states inevitably change, and hence, the mapping from states to values for a particular state can be traced by recording the realized optimal value in a trial when the environment is in that state.

In general, agent actions may affect states. In our finance example, this would mean that the investment decisions of our agent would affect the state of the econ-

omy, which makes little sense. Therefore, we will assume throughout that actions do not affect states. A more elaborate discussion of the relation between actions and states in the context of finance can be found in Chapter 1 of Bossaerts (2005).

Inspired by animal learning, and later confirmed in neural processes induced by animal learning (Schultz, Dayan, & Montague, 1997), machine learning has developed a remarkably simple, yet surprisingly robust algorithm for the agent to learn to optimize in dynamic settings. The idea is actually straightforward. Remember that the value of a state, say s, can be obtained by considering a trial when the environment is that state s. Therefore, one could take the action deemed optimal for the trial, record the immediate reward, observe the resulting state in the subsequent trial, and add the previously recorded value for that state. But our agent does not know (yet) what the true optimal action is, nor does the agent know what the true value is in the state in the subsequent trial; however, it has observed immediate rewards from taking an action in previous trials where the state was the same, and it may have an estimate of what the value is of the state in the subsequent trial for some cleverly chosen action (we will discuss which action to choose later on). The sum of the two constitutes a "cue" for the agent of the value of taking the proposed action. We refer to this as the "Q" value of the proposed action given the state of the present trial." Across trials where the same state is visited and the same action is taken, the estimated of the Q value can be updated by simple *adaptive expectations*: the agent updates the estimate using the *prediction error*. The prediction error equals the difference between the sum of the newly recorded reward and Q value given the new state (in the subsequent trial), and the old estimate of the Q value. The agent can do this across multiple trials. Trials are often arranged in *episodes*. This arrangement allows one to investigate what the agent has learned at pre-set points (episode ends).

Because the Q values are updated by means of adaptive expectations, the learning technique is referred to as *Temporal Difference Learning* or "TD Learning." The optimal value given a state is then obtained by choosing the action that maximizes the Q value for that state. For the technique to converge, i.e., for the Q values of the best action to converge, across states, to the true value function, two conditions are needed.

First, all states have to be visited sufficiently often. Indeed, even if one knew what the optimal action was in a given state, the value of that state is the *expected* sum of all future rewards. Expectations can be learned through adaptive expectations, but one needs to experience enough observations for it to converge; a law of large numbers has to apply.

The second, related, condition is that all actions are taken sufficiently often. If an action is rarely taken because it is deemed (estimated) to be sub-optimal, one may never learn that it is indeed sub-optimal; it may in fact be optimal!

Because of the second condition, *exploration* is necessary. The agent must not decide prematurely that certain actions are inferior; the agent has to explore all actions, no matter how inferior they may seem. Several exploration strategies have been proposed. Here, we will use the simplest one, namely, the *greedy strategy*. In the greedy strategy, the agent picks what it considers to be optimal (based on current estimates of the Q values) with probability $1-\epsilon$, while randomly picking any other action with probability ϵ. Here, ϵ may initially be a large number (less than 1), but it can be decreased over time, as the agent learns, to ultimately converge to zero, at which point the agent stops learning. Choice of the exploration strategy will not resolve the issues with tail risk that we study here, which is why we stick to the simplest strategy.

One more detail about TD Learning needs to be clarified. It was mentioned before that the Q value of an action-state pair equals the sum of the immediate reward from the action plus the Q value of a suitably chosen action in the subsequent trial. In our application of TD Learning, we will choose the *optimal* action in the subsequent trial. Optimality is determined by the Q values across actions given the state in that trial. This approach is referred to as "SARSA," which is short for State-Action-Reward-State-Action, indicating that only actions (deemed) optimal are taken. An alternative would be to use the Q value in the subsequent trial associated with an action which is optimal only with probability $1-\epsilon$, i.e., an action that follows the greedy exploration policy. This choice was made in the original version of "Q Learning."

In TD learning, the Q value is learned through adaptive expectations. A recently suggested alternative would be to summarize all past observed Q values in the form

of an empirical distribution (or a histogram),[6] update this distribution in every trial where the same state occurs and the same action is taken, and use the mean of this empirical distribution as new estimate of the Q value. This approach is known as *Distributional Reinforcement Learning*, abbreviated disRL. See (Bellemare et al., 2017). It has been shown to be far more effective than the recursive TD learning procedure, especially in strategic interaction (games).

DisRL provides another advantage: it allows one to introduce neural network techniques in order to determine which elements of a set of states are relevant for optimal decision-making, and how Q values relate to these elements. Indeed, in a finance context, it could be that the Treasury bill rate and the dividend yield are potential candidates for optimally switching into and out of stocks, in addition to a change in industrial production, but only one is actually relevant. Therefore, we need a technique that determines which of the three is/are relevant.

The technique that combines disRL with neural networks is referred to as *deep RL*. It is meant to simultaneously solve for the optimal action in a dynamic environment and solve the "credit assignment problem," i.e., the question as to which aspects of the environment are relevant to determine optimal actions. We will not be concerned with deep RL here.

4.2.3 *Our Contribution*

When tail risk affects the environment, i.e., when rewards are subject to frequent and large outliers, neither the recursive traditional TD learning nor the more recent Distributional RL are robust. This is because the estimate of the Q value for a state-action pair, whether obtained through adaptive expectations (i.e., recursively), or through the mean of the empirical distribution of past reward outcomes and Q estimates, is sensitive to these outliers.

At its core, the method we propose is a simple, but profound improvement of distributional RL. The idea is not to estimate a Q value using the mean of the estimated empirical distribution, but to use the most efficient estimator of the first

[6] We focus here on techniques that use the empirical distribution, since it is a consistent estimate of the true distribution, while the histogram is not consistent as an estimate of the true density.

(signed) moment. Under leptokurtosis, the mean of the empirical distribution generally exhibits low efficiency, that is, its standard error is not the lowest. If a maximum likelihood estimator of the mean exists and is asymptotically efficient, it will provide the most efficient estimator, however (technically: it will reach the Cramér-Rao lower bound). We propose to use this estimator.

The contribution is simpler to explain than to put into practice. This is because tail risk affects the two components of the TD prediction error differentially. Tail risk may always affect immediate rewards. However, eventually, after a suitably long learning period, it should no longer affect the observed Q value in the subsequent trial since Q values are the *expected* sum of future rewards. Therefore, we account differentially for the two components: we apply efficient disRL only to the reward term, implementing traditional recursive TD learning for the Q value term. How this is done technically is explained next.

Appeal to maximum likelihood estimation requires the researcher to commit to a family of distributions from which the rewards (conditional on the state-action pair) are thought to have been generated. In the case of tail risk, i.e., under leptokurtosis, the t distribution provides a good working hypothesis. This is the family we use here. Even if the true distribution is not t, the approach gives good results; one can think of the use of the t distribution as "quasi maximum likelihood:" for the purpose of TD learning, it provides desired asymptotic properties.

In the context of tail risk, the improvements can be dramatic, as we demonstrate with an experiment. It deserves emphasis that these improvements emerge even if our approach does not increase the speed of convergence as a function of sample size (number of occurrences of the state-action pair); convergence will remain inversely proportional to the square root of the sample size. But the constant of proportionality will be decreased markedly, sufficiently so that our approach becomes robust to tail risk, while traditional TD Learning and distributional RL lack robustness.

There are environments where our *efficient* version of disRL affects also the speed of learning as a function of sample size. In the Conclusion section, we provide an example where the rewards are generated by a shifted-exponential distribution. There, the estimation error of the Q values, can be reduced, not in inverse proportion to the square-root of the sample size, but in inverse proportion of the sample size.

4.3 PRELIMINARIES

4.3.1 *TD Learning*

We model the interaction of an agent with an environment in the traditional way, as a Markov Decision Process (MDP). An MDP is defined by (S, A, R, P) where S denotes the set of states, and A denotes the set of the available actions. R is a random reward function that maps each state-action pair to a random reward that lives in an outcome space F, that is, $R : S \times A \to F$.[7] $P(s^{'}|s, a)$, s, $s^{'} \in S$ and $a \in A$, denotes the state transition distribution from one trial to another. Primes ($^{'}$) are used to denote "subsequent trials." Let $\pi(a|s)$ be the policy, i.e. the probability of action a in state s.

We denote Q^{π} as the action-value function for a particular policy π when the initial ($t = 0$) state is s and the initial action is a. It is calculated as a discounted sum of expected future rewards if the agent follows the policy π from $t = 1$ onward:

$$Q^{\pi}(s, a) := \mathbb{E}_{\pi,P}[\sum_{t=0}^{\infty} \gamma^{t} R(s_t, a_t)|s_0 = s, a_0 = a]$$

where $s_t \in S, a_t \in A$ and $\gamma \in [0, 1)$ is the discount factor. At the same time, Q^{π} is a fixed point of the Bellman Operator T^{π} (R. E. Bellman, 1957),

$$T^{\pi}Q(s, a) := \mathbb{E}[R(s, a)] + \gamma \sum_{s^{'},a^{'}} P(s^{'}|s, a)\pi(a^{'}|s^{'})Q(s^{'}, a^{'})$$

The Bellman operator T^{π} is used in *policy evaluation* (Sutton & Barto, 2018). We intend to find an optimal policy π^{*}, such that $Q^{\pi^{*}}(s, a) \geq Q^{\pi}(s, a)$ for all (s, a, π). The optimal state-action value function $Q^{\pi^{*}}$ is the unique fixed point of the Bellman optimality operator T^{*},

$$T^{*}Q(s, a) := \mathbb{E}[R(s, a)] + \gamma \sum_{s^{'}} P(s^{'}|s, a) \max_{a^{'}} Q(s^{'}, a^{'})$$

[7] To simplify things, we suppress the stochastic index "ω" which is used in probability theory to capture randomness. ω only affects the rewards, and not the value function Q. This distinction will be important when we discuss de-coupling of the terms of the RL updating equations.

In general, we do not know the full model of the MDP, let alone the optimal value functions or optimal policy. Therefore, we introduce TD learning, which exploits the recursive nature of the Bellman operator (Sutton & Barto, 2018).

There are different versions of TD Learning. Watkins and Dayan (1992) proposed the following updating:

$$Q(s,a) \leftarrow Q(s,a) + \alpha(R(s,a) + \gamma \max_{a'} Q(s',a') - Q(s,a)),$$

where

$$R(s,a) + \gamma \max_{a'} Q(s',a') - Q(s,a)$$

is referred as the *prediction error*, or "TD-error," and α is the learning rate. The updating rule uses the estimate of the Q value in the subsequent trial for the action that is optimal given the state. This particular updating version of TD learning is called SARSA (State-Action-Reward-State-Action). Watkins and Dayan (1992) have shown that actions and action-state values converge to the true (optimal) quantities, provided the agent explores, and visits all states, sufficiently. To accomplish this, the agent has to ensure the learning rate α does not decrease too fast, and to use an exploration policy (choices of actions a) that tries out all possibilities sufficiently often. Here, we will use the greedy policy, whereby the agent chooses the optimal action in a state $\arg\max_a Q(s,a)$ with chance $1-\epsilon$ and a randomly chosen sub-optimal action with chance ϵ, for some $\epsilon > 0$ which is reduced over time.

4.3.2 *Distributional Reinforcement Learning (disRL)*

Similar to the Bellman operator in TD Learning, the distributional Bellman operator, T_D^π is formally defined as:

$$T_D^\pi Z(s,a) \overset{D}{=} R(s,a) + \gamma Z(s',a'), \tag{4.1}$$

where $\overset{D}{=}$ means equality in distribution (Rowland et al., 2019; Bellemare et al., 2017). States s' are drawn according to $P(s'|s,a)$, actions a' are from the policy π, and

rewards are drawn from the reward distribution corresponding to the state s and action a.

One of the proposed algorithms, the categorical approach (Bellemare et al., 2017), represents distributions in terms of histograms. It assumes a categorical (binned) form for the state-action value distribution. As such, the categorical approach approximates the value distribution using a histogram with equal-size bins. The histogram is updated in two steps: (i) shifting the entire estimated probability distribution of estimated Q values using the newly observed reward, (ii) mapping the shifted histogram back to the original range. Notice that the range is fixed beforehand. As we shall see, this is problematic especially in the context of tail risk.

In disRL, the estimated distribution (in the form of, e.g., the histogram, as in categorical disRL) is mostly not used directly to obtain an estimate of the mean Q value. Instead, the distance of the estimated distribution from an auxiliary, parametric family of distributions is minimized. This allows for flexible relationships between the Q value distributions and the states and actions. Often neural networks are fitted, whence the term "deep RL." Means of the fitted distributions are then computed by simple integration. For instance, in categorical disRL, auxiliary probabilities $p_i(s, a|\theta)$ for each bin i (given the state s and action a) are obtained by minimizing the distance between them and the histogram. The relation between the state-action pairs (s, a) and the probability of the ith bin is fit with a neural network with parameter vector θ using the (non-linear) Least Square method. The mean Q value (given the state s and action a) is then computed by integrating over the auxiliary distribution:

$$Q(s, a) = \sum_{i=1}^{K} p_i(s, a|\theta) z_i,$$

where the z_i are (evenly spaced) midpoints for the K bins of the histogram.

4.4 LEPTOKURTOSIS

In principle (and as we shall see later through simulations), TD Learning and disRL are not well-equipped to handle a leptokurtic reward environment. There are at least three reasons for this.

1. Leptokurtosis of the reward distribution. disRL simply integrates over the distribution in order to estimate the true Q value. Traditional TD Learning uses a recursive estimate. Both are inefficient under leptokurtosis. Indeed, in general there exist much better estimators than the sample average, whether calculated using the entire sample, or calculated recursively. The most efficient estimator is the one that reaches the Cramér-Rao lower bound, or if this bound is invalid, the Chapman-Robbins lower bound (Casella & Berger, 2021; Schervish, 2012). Under conditions where maximum likelihood estimation (MLE) is consistent, MLE will provide the asymptotically most efficient estimator with the lowest variance: it reaches the Cramér-Rao lower bound (Casella & Berger, 2021). Often, the MLE estimator of the mean is very different from the sample average. This is the case, among others, under leptokurtosis.

2. Heterogeneity of the prediction error. The prediction error (TD error) is the sum of two components, the period reward $R(s, a)$ and the (discounted) increment in values $\gamma Q(s^{'}, a^{'}) - Q(s, a)$. As the agent has learned the optimal policy and the state transition probabilities, Q values converge to *expectations* of sums of rewards. These expectations should eventually depend only on states and actions. Since the distribution of state transitions is generally assumed to be non-leptokurtic (e.g., Poisson), the increment in Q values $\gamma Q(s^{'}, a^{'}) - Q(s, a)$ will no longer exhibit leptokurtosis. At the same time, rewards $R(s, a)$ continue to be drawn from a leptokurtic distribution. As a result, the prediction error is a sum of a leptokurtic term and an asymptotically non-leptokurtic term. The resulting heterogeneity needs to be addressed. Measures to deal with leptokurtosis may inadversely affect the second

term.[8] The two terms have to be decoupled during updating. This is done neither in traditional TD Learning nor in disRL.

3. Non-stationarity of the distribution of Q values. As the agent learns, the empirical distribution of Q values shifts. These shifts can be dramatic, especially in a leptokurtic environment. This is problematic for implementations disRL that proceed as if the distribution of Q values is stationary. Categorical disRL, for instance, represents the distribution by means of histogram defined over a pre-set range. Outlier rewards may challenge the set range (i.e., outliers easily push the estimated Q-values beyond the set range). One could use a generous range, but this reduces the precision with which the middle of the distribution is estimated. We will illustrate this with an example when presenting results from our simulation experiments. Recursive procedures, like those used in the Kalman filter or in conventional TD learning, are preferred when a distribution is expected to change over time. Versions of disRL that fix probability levels (e.g., by fixing probability levels, as in quantile disRL; Dabney, Rowland, et al. (2018)) allow the range to flexibly adjust, and therefore can accommodate nonstationarity. These would provide viable alternatives as well.

4.5 PROPOSED SOLUTION

In this section, we first present a simple environment where leptokurtosis can easily be introduced. We subsequently propose our enhancement of disRL, aimed at imputing robustness into RL in the face of leptokurtosis.

4.5.1 *Environment*

We create a canonical experimental environment that mimics typical decision-making in financial markets. There are two states $S : \{s_0, s_1\}$, and two available actions

[8] Maximum likelihood estimation of the mean of a leptokurtic distribution, the t distribution for instance, eliminates the influence of outliers by setting them (close to) zero, as we shall document later (see Figure 4.2). The resulting estimator is less efficient than the simple sample average when the distribution is not leptokurtic, since observations are effectively discarded.

$A : \{a_0, a_1\}$. So, agents' actions have no effect on the states: $P(s^{'}|s, a) = P(s^{'}|s)$. (We continue to use primes $[^{'}]$ to denote outcomes in the subsequent trial.) The state transition probability $P(s^{'}|s)$ is such that the probability of staying in the same state is higher than the probability of switching to another state. One can view our environment as a *two-arm contextual bandit problem*, with two discrete states and uncertain rewards. A graphical depiction in terms of a finite automaton is provided in Figure 4.1.

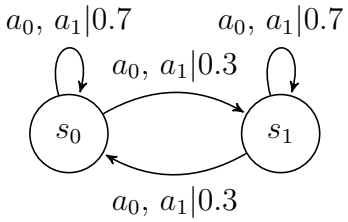


Figure 4.1. Environment, depicted in terms of a finite automaton.

We consider three different reward distributions. One is the standard Gaussian distribution; the second one is a t distribution with low degrees of freedom, and hence, high leptokurtosis. The third one is the empirical distribution of daily returns on the S&P 500 index. We comment later how the second and third distributions are related. Details are provided in Table 4.1. In all cases, the expected reward is the highest for action a_0 in state s_0 and action a_1 in state s_1. Consequently, the optimal policy is pure-strategy with optimal state-action pairs equal to (s_0, a_0) and (s_1, a_1).

The traditional Gaussian reward structure provides the benchmark for the leptokurtic environments. For the first leptokurtic case, we take a t distribution with 1.1 degrees of freedom. This distribution has been popular in finance to model leptokurtosis (Alberg, Shalit, & Yosef, 2008; Franses, Van Der Leij, & Paap, 2007; Mittnik, Paolella, & Rachev, 1998; Bollerslev et al., 1987). The third case uses the empirical distribution of daily returns on the S&P 500 index. As we shall document later, when approximated using a t distribution, the degrees of freedom are estimated to be about 3. Interestingly, the fourth moment (which tracks leptokurtosis)

	Reward Distribution $R_t(s,a)$					
	Gaussian		Leptokurtic		Empirical S&P 500	
	s_0	s_1	s_0	s_1	s_0	s_1
a_0	$\boldsymbol{N_{2.0,1}}$	$N_{1.5,1}$	$\boldsymbol{t_{2.0,1,1.1}}$	$t_{1.5,1,1.1}$	$\boldsymbol{\mu_d + 2.0}$	$\mu_d + 1.5$
a_1	$N_{1.5,1}$	$\boldsymbol{N_{2.0,1}}$	$t_{1.5,1,1.1}$	$\boldsymbol{t_{2.0,1,1.1}}$	$\mu_d + 1.5$	$\boldsymbol{\mu_d + 2.0}$

Table 4.1. Reward Distributions: Three Cases. The optimal policy (highlighted in bold) is $\pi(s_i) = a_i$ for $i \in \{0, 1\}$. $N_{l,1}$ is the normal distribution with mean l and scale (standard deviation) 1, $t_{l,1,1.1}$ is the location-scale student t-distribution with location l, scale 1, and degrees of freedom equal to 1.1, and μ_d is the empirical distribution of daily returns of the S&P 500 index.

does not exist when the number of degrees of freedom equals 4 or less. In other words, leptokurtosis is extreme, even for an index as diversified as the S&P 500.

We take the difference in mean rewards between optimal and sub-optimal actions to be equal to 0.5. In terms of returns, this implies that the difference in mean returns is a fraction of the reward standard deviation in the Gaussian case (or a fraction of the scale in the case of the t distribution). This is to best emulate experience in field financial markets: historically, the Sharpe ratio (expected reward to standard deviation ratio) of returns on financial assets tends to be below 1.0, i.e., the expected return is a fraction of the return standard deviation. See, e.g., (Bogle, 2016).

4.5.2 *Efficient disRL (e-disRL)*

The key distinguishing features in our approach are that we (i) decouple the two terms of the prediction error in TD Learning, and (ii) use efficient estimation of the mean of the first term ($R(s,a)$ in Equation 4.3.1), exploiting, as in disRL, the availability of the entire empirical distribution, while (iii) applying standard recursive estimation on the second term of the prediction error ($\gamma Q(s^{'}, a^{'}) - Q(s,a)$ in Equation 4.3.1).

To disentangle the effect of separating the two terms of the prediction error and the use of efficient estimation of the mean, we proceed in stages, and report results, first, for an estimator that only implements the separation but continues to use

the sample average as the estimator of the expected rewards, and second, for an estimator that both separates the components of the TD error and applies efficient estimation when calculating the mean of the empirical distribution of rewards. We refer to the former as "e-disRL-," and the latter as "e-disRL."

Summarizing:

1. **e-disRL-**: Rewards and discounted action-value updates are separated; standard recursive TD learning is applied to the latter, and standard disRL to the former (i.e., the mean is estimated by simple integration over the empirical reward distribution).
2. **e-disRL**: Same as e-disRL-, but we use an efficient estimator for the mean of the reward distribution.

Algorithm 4.1: Pseudo-Code for e-disRL- and e-disRL

```
1 for episode in n episodes do
2     for step t in m steps do
3         a = ε-greedy(Q(s, a));
4         R_t(s, a) = environment(a);
5         update M(s, a) by appending R_t(s, a);
6         Q(s, a) ← Q(s, a) + α(Ê(R_t(s, a)) + γQ(s', a') − Q(s, a))
7     end
8 end
```

Algorithm 4.1 specifies the updating in terms of pseudo code. The algorithms for e-disRL- and e-disRL only differ in how $\hat{E}(R_t(s,a))$ is computed. In both cases, per state-action pair, we record rewards received in a history buffer $M(s,a)$ (see line 5 in Algorithm 4.1). In e-disRL-,

$$\hat{E}(R_t(s,a)) = \frac{1}{n_t}\sum_{i=0}^{n_t} R_i(s,a)$$

where n_t is the number of rewards recorded for state-action pair (s,a), or the dynamic length of the history buffer $M(s,a)$, and $R_i(s,a)$ is individual reward in the buffer. In case of e-disRL, we employ the Maximum Likelihood Estimation principle and

obtain an estimate of the mean by applying the MLE estimator to the history buffer $M(s,a)$.

The MLE estimator of the mean of a Gaussian distribution equals the sample average, hence in the Gaussian environment there is no difference between e-disRL- and e-disRL. However, when rewards are generated by the t distribution, the MLE estimator differs. Here, we deploy the MLE estimator of the mean when the scale parameter is unknown, but the number of degrees of freedom fixed. See Liu and Rubin (1995). An analytical expression does not exist, but a common approach is to use an iterative procedure, the *Expectation-Maximization* (EM) algorithm. The algorithm first performs an Expectation step (E) on the history buffer:

$$\hat{w}_i = [(v+1)s^2]/[vs_t^2 + (R_i(s,a) - \bar{x})^2],$$

where, initially,

$$\bar{x} = \frac{1}{n_t}\sum_{i=0}^{n_t} R_i(s,a)$$

and

$$s^2 = \frac{1}{n_t}\sum_{i=0}^{n_t}(R_i(s,a) - \bar{x})^2.$$

This is followed by a Maximization step (M):

$$\hat{E}(R_t(s,a)) = \sum_{i=0}^{n_t} \hat{w}_i R_i(s,a) / \sum_{i=0}^{n_t} \hat{w}_i,$$

where v is the degrees of freedom, and n_t is the length of the history buffer $M(s,a)$ in trial t. The two steps can then be repeated until convergence, each time substituting the new estimate $\hat{E}(R_t(s,a))$ from step 2 for the old estimate of the mean in step 1 ($\bar{x}$) and recomputing the sample variance using the same weights as for the mean:

$$s^2 = \sum_{i=0}^{n_t} \hat{w}_i (R_i(s,a) - \bar{x})^2 / \sum_{i=0}^{n_t} \hat{w}_i.$$

Importantly, the MLE estimator does not simply truncate samples, as in many versions of robust estimation (e.g., Huber's approach; see Sun, Zhou, and Fan (2020)). The MLE estimator down-weighs outliers in surprising ways; see Figure 4.2.

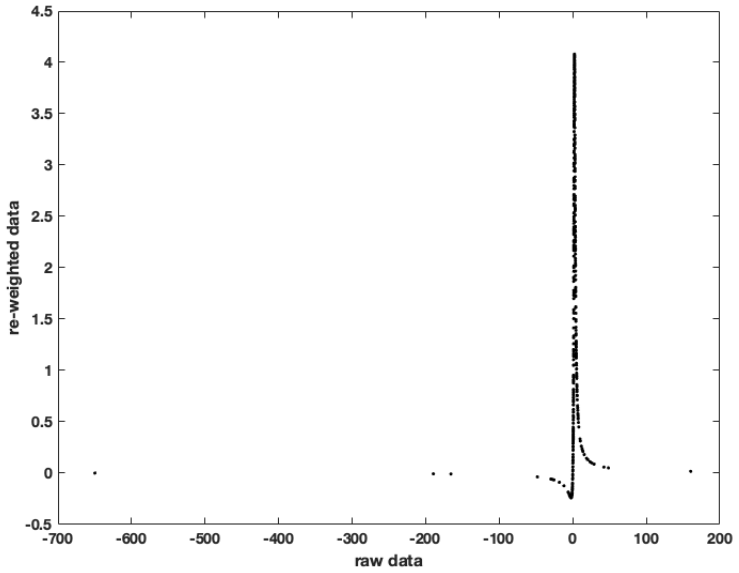


Figure 4.2. Transformation of outcomes in MLE estimation of the mean of a t distribution with location 2.0, scale 1 and 1.1 degrees of freedom. Based on a sample of 500 observations. Outcomes in the tails of the distribution are not truncated; instead, they are mapped close to zero, eliminating their impact. Outcomes close to the (estimated) mean have maximum influence; they are mapped into the highest and lowest values. The transformation is non-monotonic.

The estimator compresses the outcomes into a finite interval, and even pushes outliers closer to the mean than non-outlier outcomes. The MLE estimator equals the sample average of *non-monotonically* transformed outcomes.

4.5.3 *Convergence*

Convergence of estimated Q values to true ones is not an issue for e-disRL as it builds on a mix of disRL (for estimation of the expected immediate reward component of the prediction error) and TD learning (for the expected Q values in the state of

the subsequent trial, conditional on optimal action). The efficient estimators of the mean reward merely change speed of convergence, not convergence *per se*.[9]

4.6 SIMULATION EXPERIMENTS

4.6.1 *Methods*

To evaluate our approach, we ran TD Learning (SARSA version), categorical disRL, e-disRL- and e-disRL on the "Gaussian," "Leptokurtic," and the "Empirical S&P 500" reward settings in our canonical two-state environment. To implement disRL, we replicated the original categorical approach proposed in (Bellemare et al., 2017), with slight modifications to make it suitable for our task. The number of bins (K) in the histogram was set to 100, and the midpoints were equispaced in a range from -30 to +30. We refrained from parametrically fitting a family of distributions to the histograms because this step is used in disRL to learn an unknown relation between state-action values and states when the state space is complex. In our environment, there are only 2 discrete states, and hence, there is no need to learn the relationship Q values-states. Elimination of the parametric fitting step allows us to focus on the action-value learning. We compute the mean of the histogram by integration over the histogram.

An experiment consists of 100 game plays. Each game play contains 200 episodes, comprised each of 100 steps (trials/periods). The episodes artificially divides game play into epochs of equal length, to evaluate how much the agent has learned, and to control exploration and learning rates. Episodes 1-10 (1000 trials in total) are exploratory, whereby the agents uses the greedy algorithm with an exploration rate ε that decays exponentially from 0.9. After 10 episodes, exploration stops, and the agent implements the policy deemed optimal at the beginning of a new trial. The discount factor, $\gamma = 0.9$, and the learning rate, $\alpha = 0.1$, are both fixed throughout. We dynamically adjust the length of the buffer with historical rewards, which implies

[9]Traditional disRL approximates the empirical distribution of Q values with a parametric family of distribution, in order to capture complex relationships between a large state space and state-action values. This approximation complicates convergence proofs; see, e.g., Bellemare et al. (2017); Dabney, Ostrovski, et al. (2018); Rowland et al. (2018).

that the buffer grows across trials and episodes. Figure 4.3 displays a graphical depiction of the setup.

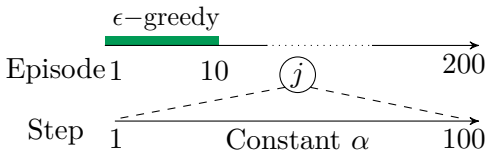


Figure 4.3. Experiment Timeline.

All the results and analyses below are conducted solely on post-exploration data. That is, they reflect actions and Q values after the 10 exploratory episodes (or $10 \times 100 = 1000$ trials).

Other parameter configurations could be envisaged. We show below, however, that the parameter choice works well in the baseline, Gaussian environment. There, we find that all learning approaches learn effectively. TD Learning and (categorical) disRL report the correct policy at the end of $\approx 50\%$ of episodes beyond episode 10. e-disRL-/e-disRL increase this percentage to 100% (in the Gaussian case, e-disRL- and e-DisRL are identical).

4.6.2 *The Gaussian Environment*

Under Gaussian rewards, traditional TD Learning generates excellent performance, but e-disRL- (and equivalently, e-disRL) is more robust, generating 100% accuracy; see Table 4.2. disRL is very disappointing, but we hasten to add that this is because we implemented the categorical version, which estimates Q values distributions using histograms. As we shall discuss at the end of this section, this requires one to commit to a finite range of possible estimated Q values. But the range of immediate rewards is unbounded, and immediate rewards have a large influence on estimated Q values in early stages of learning. As a result, bounds on the range of estimated Q values generate biases.[10]

[10]The quantiles or expectiles versions of disRL are not subject this influence. Like the empirical distribution approach we use in e-disRL, unbounded ranges are possible.

Convergence Robustness	Gaussian	Leptokurtic	Empirical S&P 500
TD	67(%)	1(%)	42(%)
disRL	5	1	1
e-disRL-	**100**	16	100
e-disRL		**80**	**100**

Table 4.2. Percentage of game plays where the artificial agent attained the optimal policy at the end of episodes 11-200. Convergence Robustness is the percentage of 100 game plays where the agent's chosen policy is the optimal one at the end of episodes 11-200. Each episode consists of 100 trials. First 10 episodes are excluded because exploration takes place. Learning is allowed throughout all episodes. Learned Q values carry over across episodes. Percentages are averaged over the two states. The Convergence Robustness of e-disRL- and e-disRL in the Gaussian case is the same, because the maximum likelihood estimator of the mean (used in e-disRL) equals the sample average (used in e-disRL-).

4.6.3 *The Leptokurtic Environment I: t-Distribution*

With leptokurtic rewards, our results confirm the importance of separately accounting for the two terms of the prediction error and using efficient estimation on the term most affected by leptokurtosis. Table 4.2 shows that both TD Learning and disRL report the optimal policy at the end of all episodes 11-100 only in 1% of game plays. e-disRL-, which simply accommodates heterogeneity of the terms of the prediction error, increases this to 16%. With efficient estimation of the mean of the reward distribution, e-disRL increases this further to an impressive 80%.

Table 4.2 may display an overly tough criterion. Table 4.3 looks at *average* performance across episodes. Standard errors show that average performance is estimated with high precision. Only e-disRL generates high levels of performance: it attains the optimal policy on average in 95%/98% of episodes, though there are game plays where it reports optimal policy in none of the episodes. The latter may be attributed to the short duration of exploration (10 episodes). The other three learning protocols report optimal policies only in slightly more than half of the episodes. Performance improves when moving from TD Learning and disRL

to e-disRL-, demonstrating that separately accounting for immediate rewards and subsequent Q values during updating is beneficial.

Learning Procedure	State	Robust Convergence	Average Convergence Mean	St Error	(Min, Max)
TD Learning	s_0	1(%)	54(%)	3(%)	(0(%), 100(%))
	s_1	1	56	3	(0, 100)
disRL	s_0	1	55	4	(0, 100)
	s_1	1	45	4	(0, 100)
e-disRL-	s_0	20	59	4	(0, 100)
	s_1	12	63	4	(0, 100)
e-disRL	s_0	**77**	**95**	2	(0, 100)
	s_1	**83**	**98**	1	(5, 100)

Table 4.3. Performance in the leptokurtic environment: Details. *Robust Convergence:* Percentage of 100 game plays the agent reports the optimal policy at the end of episodes 11-200; averages across states are reported in Table 4.2. *Average Convergence:* Percentage of episodes in a game play where optimal policy is reported at episode end. Episodes 1-10 are excluded. Mean, St Error and Min, Max are calculated over 100 game plays.

Figure 4.4 displays histograms of the prediction errors in TD Learning and e-disRL for episodes 11 through 200 during a single random game play. The distributions for TD Learning are highly leptokurtic, as they inherit the leptokurtosis of the underlying reward distribution. In contrast, the distributions of prediction errors are concentrated around zero when using e-disRL. Notice that, for e-disRL, there are hardly any prediction errors for sub-optimal actions (a_1 in state s_0 and a_0 in state 1). This is because (i) e-disRL learned the correct policy after 10 episodes, (ii) e-disRL rarely un-learned the optimal policy afterwards, (iii) the agent no longer explored (i.e., the agent always chooses the optimal action).

Figure 4.5 demonstrates that the leptokurtosis of the prediction errors for the TD Learning agent adversely impacts the estimates of the Q values. Shown are distributions of estimated Q values at the end of 200 episodes across 100 game plays. The impact is still noticeable when we merely split the prediction error and use the traditional sample average to estimate the expected reward in the immediate trial, as is the case for e-disRL-.

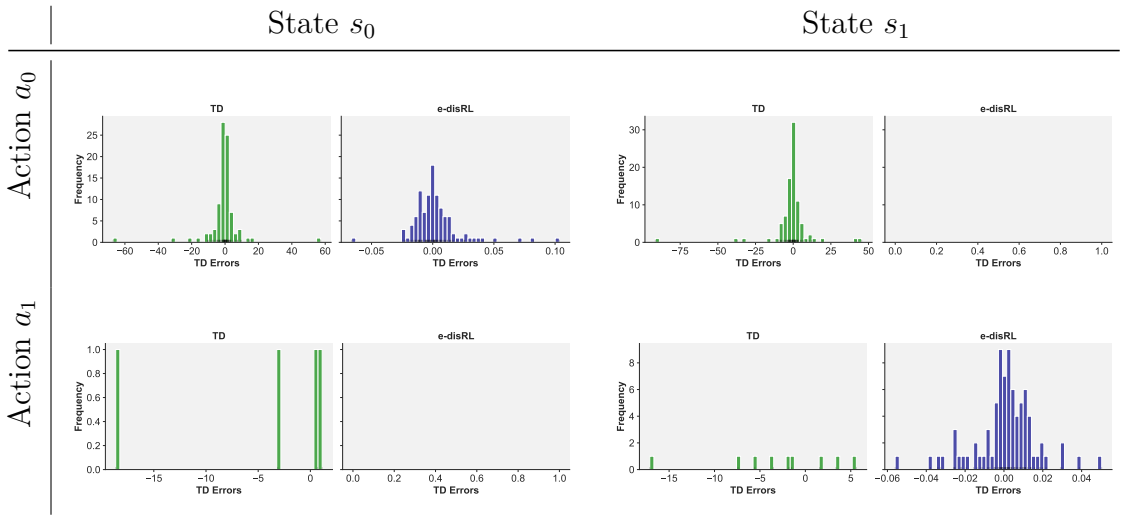


Figure 4.4. Prediction error histograms for all trials during episodes 11-200 of a single game play, TD Learning and e-disRL. There are no observations, and hence, no histograms, for e-disRL in state-action pair (s_0, a_1), because the artificial agents reached the optimal policy at the end of the 10th episode, never switched policy afterwards, and stopped exploring. In one single trial, the e-disRL agent implemented the sub-optimal policy in state-action pair (s_1, a_0).

Because it uses an efficient estimator, e-disRL produces symmetric, concentrated distributions around the true values for the optimal state-action pairs (20).[11] In the case of sub-optimal state-action pairs, the averages are below the true values (19.5).[12] This can be attributed to the fact that the e-disRL agent rarely chooses sub-optimal actions after the exploration epoch (episodes 1-10), as discussed before. As a result, the estimated Q values are rarely updated beyond episode 10. Technically, they are based on erroneously chosen actions in the calculation of the estimated Q values in the subsequent trial. The actions are chosen erroneously because the agent has not

[11]The Q values of the optimal state-action pairs can readily be computed by taking the infinite sum of maximal expected rewards (2.0) discounted with a discount factor equal to 0.9. That is, the Q values equal $2.0/(1-0.9) = 20$.

[12]For suboptimal state-action pairs, the Q values equal the immediate expected reward from a sub-optimal action, namely, 1.5, plus the expected infinite sum of discounted rewards when switching to the optimal policy in the subsequent trial and beyond. That is, the Q value equals $1.5+(0.9)(20) = 19.5$.

learned yet to identify the optimal policy. The true Q values (19.5) are instead based on the true expected reward in the immediate reward and the expected (discounted) Q value across possible states in the subsequent trial, evaluated at the truly optimal actions.[13] Incomplete learning also explains the higher variance of the estimated Q values in the case of suboptimal state-action pairs.

The distributions of Q values are highly leptokurtic for TD Learning and e-disRL-, and in the case of TD Learning, significantly left-skewed as well.

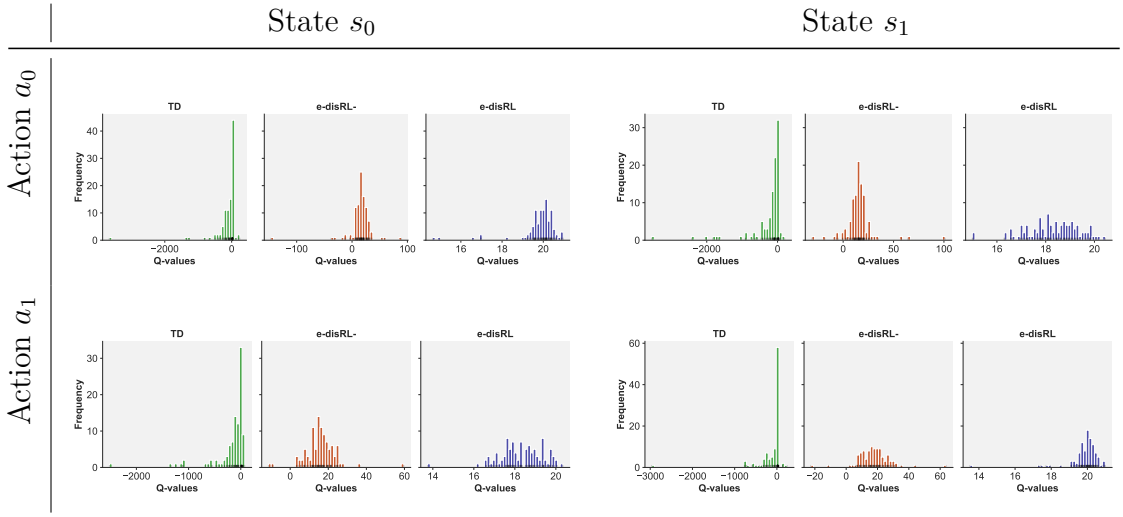


Figure 4.5. Histograms of estimated Q values at the end of episode 200 in 100 game plays. TD Learning, e-disRL- and e-disRL. The effect of tail risk is marked under TD Learning, and is still quite noticeable when decoupling the immediate reward in the prediction error while estimating its mean using the sample average of past rewards (e-disRL-). Only when an efficient estimator of the mean is used (e-disRL) does the effect of tail risk disappear.

4.6.4 *The Leptokurtic Environment II: Drawing Rewards From The Empirical Distribution of S&P 500 Daily Returns*

We now use the empirical distribution of daily open-to-close returns of the S&P500 index over the period of 1970-2019 as our reward distribution. We fit a t distribution

[13] See previous footnote for calculations.

with MLE to recover the degrees of freedom v and set the scaling factor equal to 1. We then use the fitted v to implement efficient estimation of the mean reward in e-disRL. The estimate of the degrees of freedom equalled 3.29. This is higher than the degrees of freedom we used in the second experiment, but still low enough for fourth moments not to exist, and hence, leptokurtosis (tail risk) to be extreme.

When compared TD Learning and disRL, we again record a substantial improvement when compared to e-disRL. See Table 4.4. Evidently, most of the improvement appears to emerge because of decoupling of the immediate reward and the Q value in the subsequent trial: e-disRL- reaches the same convergence statistics as e-disRL. That is, in the case of a broadly diversified index such as S&P 500, most of the issues with tail risk disappear by merely properly accounting for heterogeneity in the terms of the prediction error.

Learning Procedure	State	Robust Convergence	Average Convergence Mean	St Error	(Min, Max)
TD Learning	s_0	37(%)	99(%)	< 0.5(%)	(95(%), 100(%))
	s_1	47	99	< 0.5	(97, 100)
disRL	s_0	1	69	3	(3, 100)
	s_1	1	62	3	(1, 100)
e-disRL-	s_0	100	100	0	(100, 100)
	s_1	100	100	0	(100, 100)
e-disRL	s_0	**100**	**100**	0	(100, 100)
	s_1	**100**	**100**	0	(100, 100)

Table 4.4. Performance against the S&P 500 daily return distribution: Details. *Robust Convergence:* Percentage of 100 game plays the agent reports the optimal policy at the end of episodes 11-200; averages across states are reported in Table 4.2. *Average Convergence:* Percentage of episodes in a game play where optimal policy is reported at episode end. Episodes 1-10 are excluded. Mean, St Error and Min, Max are calculated over 100 game plays.

4.6.5 *Impact of outlier risk on categorical distributional RL*

The categorical version of disRL does not perform well, even in the baseline, Gaussian case. We attribute this to the agent's setting of the range of potential Q

values to a predetermined, fixed (and finite) range. We had set the range equal to $[-30, +30]$, based on knowledge of the optimal Q values (20 and 19.5 for optimal and sub-optimal state-action pairs, respectively) and of the reward distributions (mean 2 or 1.5). (In real-world implementation of disRL, this information may not be available!) However, in all treatments, the range of estimated Q values is unbounded since the immediate reward distribution is unbounded, so estimates of Q are unbounded too.

Figure 4.6 illustrates how range constraints adversely affects inference in the leptokurtic treatment (t distribution). The figure shows that, at the end of episode 16 of the game play at hand, 100% of the distribution is assigned to the lowest bin in state-action pairs (s_0, a_1) and (s_1, a_0). The issue is resolved in the subsequent episode (17), i.e., after an additional 100 trials, in the case of (s_1, a_0). However, it is not resolved for (s_0, a_1) and emerges anew in state-action pair (s_1, a_1). Intuitively, the results are to be expected because there is a high chance of an exceptionally large reward under a leptokurtic distribution such as that of daily returns on the S&P 500.

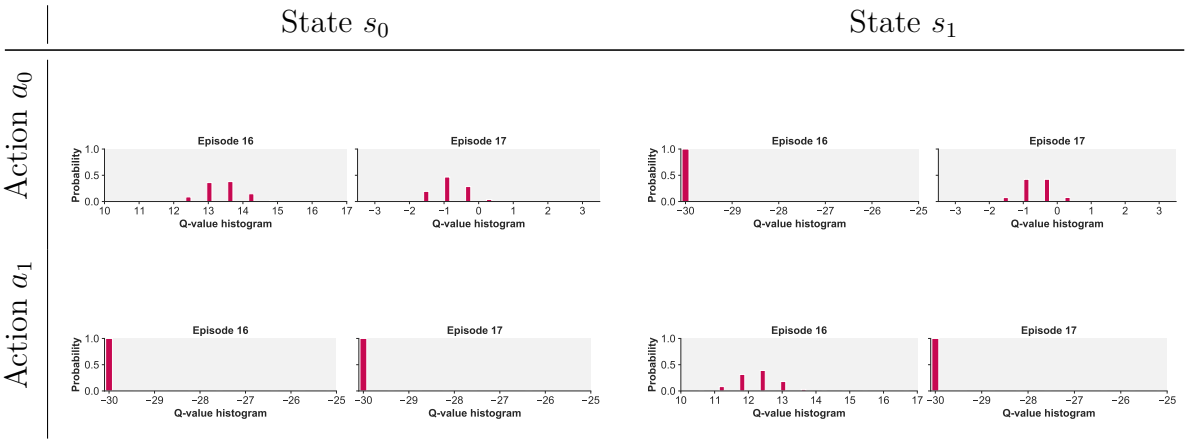


Figure 4.6. Q value histograms for (Categorical) disRL, leptokurtic case. Shown are distributions at the end of Episodes 16 and 17 in one particular game play. Game play is chosen to highlight the impact of tail risk on categorical disRL. The distributions for Episode 17 are obtained from those in Episode 16 plus 100 trials of disRL learning (without exploration). Note that the scale of the horizontal axes is not uniform across state-action pairs and episodes.

As mentioned before, versions of disRL that do not fix the range of possible estimated Q values are immune to the negative influence of tail risk displayed in Figure 4.6. These include disRL techniques based on quantiles or expectiles. This does not mean that the alternatives fully accommodate tail risk. Rowland et al. (2019) shows how even quantile-based disRL cannot deal with tail risk in the shifted-exponential distribution. A disRL approach based on expectiles works better.[14]

To resolve this issue, the authors propose to use expectiles (ER-DQN), which is another semi-non-parametric distributional method based on truncated means. Expectiles are known as expected shortfall/Conditional Value at Risk (CVaR) in finance. A CVaR is the expected loss in the tails of a distribution (Rockafellar & Uryasev, 2000); it is one method to model "tail risk". However, we show that there is a more efficient way to consider "tail risk" when estimating mean action-values.

In a shifted exponential distribution, the range depends on the same parameter that modulates the mean, which implies that the Cramér-Rao lower bound does not provide the lower bound to standard errors (Casella & Berger, 2021). Instead, the Chapman-Robbins lower bound applies; it can be reached using a simple order statistic (Schervish, 2012). We study its properties, and compare it with expectile approach, in a follow-up paper.

In the latest distributional RL study, the authors illustrate how "tail risk" could be problematic (Rowland et al., 2019). In a simple 5-state MDP where the rewards are distributed in a shifted exponential distribution, both categorical and quantile approaches fail because they do not capture the tail behavior of the distribution. To resolve this issue, the authors propose to use expectiles (ER-DQN), which is another semi-non-parametric distributional method based on truncated means. [15]
In a shifted exponential distribution, the range depends on the same parameter that modulates the mean, which implies that the Cramér-Rao lower bound does not provide the lower bound to standard errors (Casella & Berger, 2021). Instead, the Chapman-Robbins lower bound applies; it can be reached using a simple order

[14]Expectiles are related to expected shortfall/Conditional Value at Risk (CVaR) in finance. CVaR is the expected loss in the tails of a distribution (Rockafellar & Uryasev, 2000).

[15]Expectiles are known as expected shortfall/Conditional Value at Risk (CVaR) in finance. A CVaR is the expected loss in the tails of a distribution (Rockafellar & Uryasev, 2000); it is one method to model "tail risk". However, we show that there is a more efficient way to consider "tail risk" when estimating mean action-values.

statistic (Schervish, 2012). We study its properties, and compare it with expectile approach, in a follow-up paper.

4.7 CONCLUSION

Distributional RL improves on traditional RL by considering the entire distribution of action-values instead of just the mean. In a leptokurtic environment, availability of the distribution can be exploited to estimate the mean in a more efficient way, ensuring that outliers do not cause policy non-convergence or policy instability. In addition, when tail risk affects the reward distribution and not the state transitions, it is beneficial to decouple the two terms in the prediction error of RL, and apply efficient estimation of the mean only to the immediate reward component. These two considerations are the essence of our proposal, e-disRL.

In a simple, canonical investment problem, a contextual two-arm bandit problem, we demonstrated how e-disRL improves machine learning dramatically. We illustrated the importance of both efficient estimation and decoupling of the components of the RL updating equations.

From a broader perspective, our results underscore the importance of bringing prior domain-specific knowledge to machine learning algorithms using the tools of mathematical statistics. Leptokurtosis is a specific property of financial data. Mathematical statistics has developed many useful principles and tools to tackle other problems as well. Distributional RL provides the appropriate setting to introduce those principles and tools. Performance is thereby improved substantially.

An example beyond finance is an environment where rewards are generated by an exponential distribution with unknown lower bound (the so-called shifted-exponential distribution; see (Rowland et al., 2019)). There, the sample average converges to the true mean at a speed equal to the square-root of the sample size. There exists an alternative estimator of the mean, computed from the minimum observed in the sample. This estimator converges much faster, at a rate equal to the

sample size. The estimator is the most efficient one; it reaches the Chapman-Robbins lower bound on the variance of any estimator.[16]

Mathematical statistics can be brought to bear on other widely encountered problems in reward distributions, such as skewness (another feature of financial data), or small distributional shifts that become significant only after substantial accumulation ("black swans"). In each domain, distributional reinforcement learning can be exploited to obtain the most efficient way to estimate mean action-values across states, and hence, enhance control. In each case, however, it will be important to determine whether disRL should be applied to the entire prediction error, or whether one should de-couple immediate rewards from estimates of subsequent Q values.

[16]The Cramèr-Rao lower bound does not exist for the shifted-exponential distribution, which is why we refer here to the Chapman-Robbins bound.

Part IV

HUMAN ESTIMATION EFFICIENCY UNDER TAIL RISK

5

HUMAN ESTIMATION EFFICIENCY UNDER TAIL RISK

5.1 INTRODUCTION

Because humans navigate an uncertain environment, accurate estimation of expected outcomes is primordial for adapted decision-making. In finance, an investor needs to estimate monthly stock returns before deciding to buy. In health management, hospital capacity planning during a COVID epidemic requires knowledge of expected secondary infection rates. In manufacturing, a manager has to estimate mean production times before promising delivery dates.

Decisions often have to be made with limited data. Statistical efficiency is therefore called for, with the aim of extracting maximal information from the available sample. Various ways can be used to estimate expected outcomes. One approach, popular in finance, is to just take the midpoint. This would work well for monthly stock returns, but even in that case there exists a better procedure, which is to compute the sample average.[1] In technical terms, the estimator will extract most "information" (in the sense of Fisher (Casella & Berger, 2021; Schervish, 2012)) from the available sample; it will reach the "Cramèr Rao lower bound" on the standard error (Cramér, 2016; Fréchet, 1943; Darmois, 1945; Rao, 1992).

In the COVID case, a much more efficient estimator exists, because infection rates follow a heavy-tailed distribution, meaning that a small number of cases ("super spreaders") cause infections to an outsized number of people (Wong & Collins,

[1] Monthly stock returns are generally symmetric, so the midpoint is an alternative to the median as estimator of central tendency. The standard error of the median is, however, larger than that of the sample average.

2020). That is, COVID spread obeys the "Pareto principle." Increase in efficiency is obtained by cleverly downweighting the outliers and computing the sample average on the re-weighted sample. In the third case, an estimator based on the minimum observed time-till-completion vastly enhances efficiency. For a sample with, say, 25 observations, the order of magnitude of the standard error will be only 1/5th that of the sample average.[2]

If humans are well adapted to stochastic environments, it is expected that their estimation reflects concern for statistical efficiency. Here, we test this proposition. We consider three treatments. In Treatment G, the outcomes are Gaussian and hence the sample average is (one of) the most efficient estimator(s). In Treatment T, the outcomes are heavy-tailed as in COVID secondary infection counts (we use a t distribution with low degrees of freedom). In Treatment E, the outcomes are drawn from a shifted exponential distribution, thus mimicking the manufacturing case. See Fig 5.1A for histograms of typical samples that participants experienced in each of the treatments; Fig 5.1B displays corresponding sample paths of the sample average and the efficient estimator.

In all treatments, we use as baseline the midpoint estimator of the expected outcomes, for three reasons. First, this estimator is popular in practice (notably, in the finance industry). Second, it is also very parsimonious computationally: it is the average of the minimum and maximum observed outcome. Lastly, our experimental interface may have biased participants towards this estimator because participants had to indicate their estimate of the expected outcome using a slider that was initially positioned randomly in a range centered at the midpoint of the most recently observed sample (except if the previous estimate fell outside this range). See Fig 5.1C. By clicking in the middle, the participant automatically chooses the range midpoint estimate.

Animal learning, including human learning, has been widely investigated based on a theoretical paradigm called *reinforcement learning*. There, the agent learns expected outcomes using a recursive computation of the sample average. The paradigm

[2]Manufacturing completion times can be modeled with the shifted-exponential distribution. In this case, the Cramèr-Rao lower bound on standard errors is invalid. An estimator that only uses the minimum observed time reaches the alternative Chapman-Robbins lower bound. See (Hammersley, 1950; Chapman, Robbins, et al., 1951).

has been hugely successful. To our knowledge it has however never been directly tested whether humans actually use the sample average to form expectations, but instead apply more efficient procedures when available, or even resort to inefficient but computationally parsimonious estimation. Our work addresses this lacuna.

Following an earlier study on the neurobiology of Bayesian updating (d'Acremont, Schultz, & Bossaerts, 2013), we designed a probabilistic task where human participants were required to report periodically their estimation of the expected reward on one option (while the other option remained fixed) after sampling possible rewards five times (options were referred to as "bandits" – see Methods). Upon conclusion of the learning epochs, participants were asked to indicate whether they would prefer to bet on the true mean of the random reward option or the alternative risk-free option.

Participants were compensated for the accuracy of their estimates, as well as a few randomly picked choices between the random reward and the risk-free options. The latter allowed us to meaningfully confirm consistency between reported reward estimates and subsequent choices. Learning and choice were repeated twice for each treatment (G, T, E), for a total of 6 sessions. Fig 5.1D displays the timeline of the experiment.

In evaluating their estimates, we ran a mixed-effects linear regression model explaining the deviation of participants' estimates from the midpoint estimate, using as independent variables, the deviation of the sample average from the midpoint estimate, and in Treatments T and E, orthogonalized deviations of efficient estimates from the midpoint. We separated participants into two different groups, efficient and non-efficient, based on individual estimated loadings on the efficient estimator in the two replications of the treatment at hand. We then analyzed behavioral differences between the two groups, such as reaction times and consistency between submitted estimates (of the expected rewards on the risky option) and choice (between the risky and risk-free options). See Methods for further details.

5.2 METHODS

5.2.1 *Behavioral task design*

The experiment concerned pairs of gaming machines ("bandits"), one of which was *risky* because it generated stochastic outcomes, while the other was *risk-free* because it generated only a fixed reward. Participants were asked to perform two tasks: (i) to periodically estimate the mean of the risky gaming machine, after which (ii) to choose to play the risky gaming machine and be compensated by its true mean reward, or the risk-free one.

The experiment consisted of two replications of three Treatments. In each replications, displayed values of the risky gaming machine were drawn from a specific stationary distribution. At the beginning of a replication, participants were informed which distribution would apply via a short description (see Fig 5.2(a)(d)):

- Type I game-machines could generate extremely large rewards (outliers), both negative and positive.
- Type II game-machines had rewards that were bounded from above (there existed a maximum reward).
- Type III game-machines were "normal" in the sense that they rarely generated outliers; still, rewards were unbounded. They were said to "generate outcomes that looked like average daily temperatures."

The above descriptions were the only information about the stochastic nature of the risky gaming machines provided to the participants. Here we define their true data generating process (DGP):

- In the *Gaussian environment* (type III, Treatment G), the risky gaming machine values r_t were drawn independently and identically from a stationary Gaussian distribution that had a true mean μ and a standard deviation (SD) $= 1$:

$$r_t \sim N(\mu, 1) \tag{5.1}$$

- In the *Student-t environment* (type I, Treatment T), the risky gaming machine values r_t were drawn independently and identically from a stationary student-t

distribution that had a degree of freedom (dof) $v = 2.1$ and an expected value of μ:

$$r_t \sim \mu + \text{student-t}(v = 2.1) \tag{5.2}$$

- In the *Exponential environment* (type II, Treatment E), the risky gaming machine values r_t were drawn independently and identically from a stationary shifted exponential distribution that had $\lambda = 1$ and an expected value of μ:

$$r_t \sim 1 + \mu - \text{exponential}(\lambda = 1) \tag{5.3}$$

The value of the risk-free gaming machine was equal to the true mean of its paired risky gaming machine plus or minus 1.0 with 50% probability:

$$r_f = r_t \pm 1 \tag{5.4}$$

For each Treatment, there were two replications referred to as *Parts*. In each part, participants were required to conduct three tasks: a *sampling task*, an *estimation task* and a *choice task*. Each Part contained twenty episodes of the *sampling task* and the *estimation task* , and one *choice task*.

Sampling task. In each episode t, participants first observed five samples from the risky gaming machine (Fig 5.2(b)) by clicking with their mouse for five times. The true mean of the risk-free gaming machine was not displayed and participants could not interact with the risk-free gaming machine. See Fig 5.2(b).

Estimation task. Participants provided their estimation of the true mean based on all samples observed in an episode and all prior episodes in a session (Fig 5.2(c)). Estimates were indicated with a slider. Setting of the range of the slider took into account experience from prior studies, which have shown that humans exhibit excessive volatility in their estimations. Human estimates often go beyond their past observations whether explicitly instructed not to do so(Nursimulu & Bossaerts, 2014) or not(Bordalo, Gennaioli, Ma, & Shleifer, 2020). One possible reason for excessively volatile estimation is working memory constraints(da Silveira, Sung, &

Woodford, 2020). To control for excessive volatility, we restricted the range of the slider, and hence the range of possible estimates of the true sample mean. Inspired by an earlier study on human Reinforcement Learning(Nassar, Wilson, Heasly, & Gold, 2010), the range was based on a participant's estimate in the previous episode, as well as outcomes within the episode. Formally, the range of the slider $[S_{min}, S_{max}]$ was determined as follows:

- In episode $t = 1$:

$$S_{min,1} = \min\{x_j | j \in [1..5]\} \tag{5.5}$$
$$S_{max,1} = \max\{x_j | j \in [1..5]\} \tag{5.6}$$

- From episode 2, $\forall t \geq 2$:

$$S_{min,t} = S_{min,t-1} + \min(0, \min(Dist_{t-1})) \tag{5.7}$$
$$S_{max,t} = S_{max,t-1} + \max(0, \max(Dist_{t-1})) \tag{5.8}$$

where x_j are the sample values in episode t; $Dist_{t-1} = \{x_j - estimation_{t-1} | \forall j \in [1..5]\}$, and $[a..b] = \{j \in \mathbb{Z} \cap j \in [a, b]\}$.

From the second ($t = 2$) episode on, the estimate submitted in the previous episode ($t - 1$) was displayed with a green square on the slider. See Fig 5.2(c).

Choice task. Participants were asked to make a choice between the risky and the risk-free gaming machines. At this point, the reward from choosing the risk-free gaming machine was revealed. See Fig 5.2(f).

Participants subsequently moved to Part 2. See Fig 5.2(e). The locations of the risky gaming machine and the risk-free gaming machine swapped. Additionally, the mean value of both gaming machines changed. However, the distribution type of the risky gaming machine did **not** change. After Part 2, participants moved to Part 1 of a new Treatment, where the distribution of the risky gaming machine did change. See Fig 5.2(d). The change was highlighted, among others in a change of the background color. Across all treatments/replications, the distributional assignments

and the background colors were randomized in a controlled way (see supplementary information Table A.9 and A.10 for details).

Overall, the goal of the participants was to (1) minimize the average absolute prediction error $|PE|$ between the true mean and their estimations across all episodes, and (2) choose the gaming machine with a higher mean in the choice task. At the end of the experiment, participants received a monetary reward that was based on a combination of (1) and (2).

5.2.2 *Procedures*

The study has been approved by the Office of Research Ethics and Integrity at the University of Melbourne (ethics ID: 2056623.1). All participants volunteered to participate in the experiment. We conducted a power analysis to predetermine the sample size.

Forty-six (n=46) healthy young adults (mean age=22.36 years, SD=2.10 years; twenty-five females, twenty-one males) participated in our behavioral experiment. All participants were students from the University of Melbourne. Participants were provided with a plain language statement and an informed consent. *Due to Covid-19 restrictions, all sessions were conducted online.* Among others, all forms were provided online, via Qualtrics (https://lms.unimelb.edu.au/learning-technologies/qualtrics). Participants provided their written consent by typing their full name and the date in the online informed consent form.

All online communications were conducted using Zoom (http://zoom.us) due to Covid-19 lockdowns. Participants were informed that they were allowed, but not required, to use video camera. The instructor's camera remained open throughout the experiment. All participants gave verbal consent to share their browser screen when this facilitated help. Prior to screen sharing, participants were instructed to hide any sensitive information stored in their browser, for example bookmarks.

At the beginning of the experiment, an online instruction was provided to the participants. Then the instruction was screen-shared and read by the instructor. The instruction informed the participants that they would perform both the estimation task and the decision-making task. The two tasks were clearly explained

in the instructions. Participants were also told the performance evaluation and the payment methods, including the fact that the final payment was subject to their performances in both tasks. Participants were given opportunities to ask questions during and after the instruction read-out. Subsequently, participants were asked to share their screen and complete a practice experiment consisting of one complete session (Gaussian environment, two parts, each of which consisted of three sampling & estimation tasks, and one choice task). The practice session last ∼10-15 minutes. All participants acknowledged that they understood the experiment and the required tasks after the practice. After a short break, the main experiment started. The experiment was self-paced, with no time restriction. On average, a typical experiment took ∼60-90 minutes. Zoom screen share was required throughout the main experiment.

Forty-four (n=44) valid experiment data were collected. The remaining two were incomplete due to internet connection issues, and hence discarded.

5.2.3 *Analysis framework*

We implemented a standard mixed-effects linear regression model predicting the deviation of participants' estimates from the reference point and including as independent variables the deviation of efficient estimates from the reference point. Random effects on intercept and slopes were introduced at the level of participants and experimental replications ($44 * 2 = 88$ levels). Based on the random effect estimations, we separated the participants into two groups, the *Efficient* group and the *Non-Efficient* group (see main text). We compared the differences between the two groups, in terms of performance, choice, and reaction times. A summary of formal notations used in the main text is presented in the supplementary information Table A.2.

Slider midpoint as the reference. Since we chose to bias participants with adjusted minimum and maximum slider values to avoid excessive volatility in estimates, we took as reference point for the estimations the midpoint of the slider range. Preliminary analysis justified this choice: the midpoint of the slider ex-

plained a large fraction of participants' estimates. This finding can be explained by minimal-effort learning: the midpoint provides a relatively "reasonable" estimate in an uncertain environment without spending too much cognitive and motor effort. Moreover, in the finance industry, traders often adopt a "midpoint" strategy to estimate future stock price levels (see https://www.bloomberg.com/professional/blog/mid-point-fairer-price/).

In our statistical analysis of submitted estimates, we used the midpoint of the range of the slider as the reference point, and adjusted both dependent and independent variable s for it.

The Gaussian environment (Treatment G). For a Gaussian distribution, the sample average is (one of) the most efficient estimator(s) of the true mean. Hence, we tested if the deviations of participants' estimates from the midpoint were correlated with deviations of the sample average from the midpoint. Given episode $t \in [1..20]$ and participant $i \in [1..44]$, we ran the following mixed-effects linear regression, in Wilkinson notation:

$$DP_{i,t} \sim DA_{i,t} + (1 + DA_{i,t}|i : R), \tag{5.9}$$

where $DP_{i,t}$ is the difference between participant i's estimation $P_{i,t}$ and the midpoint value of the minimum and the maximum value of the slider $M_{i,t}$ in episode t, namely,

$$DP_{i,t} = P_{i,t} - M_{i,t}, \tag{5.10}$$

and $DA_{i,t}$ is the difference between the midpoint $M_{i,t}$ and the running sample average $SA_{i,t}$ of all values observed by participant i up to episode t, namely,

$$DA_{i,t} = SA_{i,t} - M_{i,t}, \tag{5.11}$$

$$SA_{i,t} = \frac{1}{N_{i,t}} \sum_{j=0}^{N_{i,t}} x_j, \tag{5.12}$$

where $x_j \in X_{i,t} = \{v_j | j \in [0..5t]\}$ for each participant i and $N_{i,t} = |X_{i,t}|$. The notation "$(1 + DA_{i,t}|i : R)$" implies that random effects were allowed for at the level of individual (i) and session/replication (R), for both the intercept and the slope.

The model was selected based on a comprehensive search using Akaike and Bayesian Information Criteria.

Orthogonal regression residuals, Student-t and exponential treatments. For the Student-t and the exponential distributions, we defined $DE_{i,t}$ as the distance between the midpoint $M_{i,t}$ and the running efficient estimator $EE_{i,t}$ of all values received by participant i up to episode t:

$$DE_{i,t} = EE_{i,t} - M_{i,t}. \tag{5.13}$$

The exact form of the efficient estimator $EE_{i,t}$ depends on the distribution. However, it is important to note that by definition, both the sample average estimates and the efficient estimates asymptotically converge to the true mean. As a result, $DA_{i,t}$ and $DE_{i,t}$ tend to be correlated. To mitigate multi-colinearity, we orthogonalized the latter relative to the former, as follows. We first ran the following linear regression for each participant i (in Wilkinson notation):

$$DE_t \sim 1 + DA_t, \tag{5.14}$$

then obtained the orthogonal residual values and used them as regressor in the mixed-effects linear regression model:

$$OR_t = DE_t - \hat{DE}_t. \tag{5.15}$$

As a result, the regressor $OR_{i,t}$ captured the *additional efficiency* in participant estimates, on top of that of the sample average.

The Student-t environment (Treatment T). The most efficient estimator is an iterative MLE procedure, the *Expectation-Maximization (EM)* model. This estimator is the most efficient estimator because it reaches the "Cramér-Rao lower bound"(Casella & Berger, 2021). Here we adopted the approach where the scale parameter is unknown, but the degrees of freedom is known(Liu & Rubin, 1995).

The approach first performs an Expectation (E) step on the values received up to episode t:

$$\hat{w_j} = [(v+1)s_{i,t}^2]/[vs_{i,t}^2 + (x_j - SA_{i,t})^2]. \tag{5.16}$$

We assumed that the degree of freedom $v = 2.1$ is known. $SA_{i,t}$ is the sample average estimates, $x_j \in X_{i,t} = \{v_j | j \in [0..5t]\}$ for each participant i, $N_{i,t} = |X_{i,t}|$ is the sample size and

$$s_{i,t}^2 = \frac{1}{N_{i,t}} \sum_{j=0}^{N_{i,t}} (x_j - SA_{i,t})^2. \tag{5.17}$$

This is followed by a Maximization step (M):

$$EE_{i,t} = \sum_{j=0}^{N_{i,t}} \hat{w_j} x_j / \sum_{j=0}^{N_{i,t}} w_j. \tag{5.18}$$

Notice that our procedure is a variation of the full EM-MLE procedure. A full EM-MLE algorithm repeats the above EM iteration and replaces $SA_{i,t}$ with the $t-1$ mean estimates until convergence given a fixed sample size. Our algorithm effectively performs only one iteration given a fixed sample size. The reasons are twofold. Firstly, asymptotically $SA_{i,t}$ has minimal influence on weight $\hat{w_j}$ since $SA_{i,t}$ is consistent. Secondly, one iteration is more biologically plausible since iteration till convergence requires trimming the outliers multiple times in a single estimation task, a cognitively extremely demanding task.

We then ran the following mixed-effects linear regression (again in Wilkinson notation):

$$DP_{i,t} \sim DA_{i,t} + OR_{i,t} + (1 + DA_{i,t} + OR_{i,t} | i : R). \tag{5.19}$$

The model was selected based on a comprehensive search using Akaike and Bayesian Information Criteria.

The Exponential environment (Treatment E). Unlike in the Student-t environment, the "Cramér-Rao lower bound" does not exist in the Exponential environment because the range of an exponential distribution depends on the same parameter that modulates the mean(Casella & Berger, 2021). Instead, an alterna-

Lags	AIC	BIC	LL
0	125.00	180.02	-52.90
1	66.14	147.47	-18.07
2	-13.38	99.34	27.89
3	-74.62	74.08	65.31

Table 5.1. Model selection criteria for Treatment E. Lags: number of lags imposed on $DE_{i,t}$. AIC: Akaike Information Criteria. BIC: Bayesian Information Criteria. LL: Log Likelihood.

tive bound exists, the "Chapman-Robbins lower bound"(Casella & Berger, 2021; Schervish, 2012). One of the estimators that reaches this bound uses the running maximum:

$$EE_{i,t} = \max\{X_{i,t}\} + \frac{1}{N_{i,t}} - 1, \tag{5.20}$$

where $X_{i,t} = \{v_j | j \in [0..5t]\}$ for each participant i and $N_{i,t} = |X_{i,t}|$.

The model selection criteria from Table 5.1 had suggested including more lags of $DE_{i,t}$ in the mixed-effects linear regression. When we added more than two lags, the coefficients were close to zero and insignificant (coefficient of $DE_{i,t-3} = 0.004$ with a p-value of 0.66). Hence, we chose to add two lags of $DE_{i,t}$ in the regression:

$$DP_{i,t} \sim DA_{i,t} + OR_{i,t} + DE_{i,t-1} + DE_{i,t-2} + (1 + DA_{i,t} + OR_{i,t} + DE_{i,t-1} + DE_{i,t-2} | i : R) \tag{5.21}$$

The model was selected based on a comprehensive search using Akaike and Bayesian Information Criteria (see Table 5.1).

Efficient vs. non-efficient group. In the Gaussian environment, we defined the efficient group to be those participants who had positive random-effect coefficients on $DA_{i,t}$ in *both* replications (the initial Part 1 and the repetition Part 2). In the Student-t and the Exponential treatments, we defined the efficient group to be those participants who had positive random-effect coefficients on $OR_{i,t}$, again in both replications. The non-efficient group consists of the remaining participants. We emphasize that the efficient group is treatment sensitive; that is, those who were

efficient in one environment were not necessarily efficient in another environment.

Choice consistency. We considered participants' choice between the risky and the risk-free gaming machine to be *consistent* with their estimation if they chose the risky (risk-free) gaming machine when their final estimation value was higher (lower) than the value of the corresponding risk-free choice. Otherwise, their choice was deemed *inconsistent* with their final estimation. Note here we disregarded "accuracy", that is, whether the selected choice was in fact the gaming machine with higher mean.

Response times. We recorded participants' response times (RT) between the action in a previous task and the action in the current task. For example, participants' choice RT is defined as the time span between when they "clicked" the button to finish the final estimation task and when they "clicked" the button to finish the choice task.

5.3 RESULTS

We denote i as each individual participant; t as each episode where participants were required to sample five values and report one estimation; $DP_{i,t}$ as the difference of participants' estimates and the midpoint (of the range of sampled values in the epoch; if the participant's estimate in the prior epoch was outside this range, the range was extended to include this prior estimate); $DA_{i,t}$ as the difference between the sample average computed from the cumulative samples since the beginning of the session and the range midpoint; $DE_{i,t}$ as the difference of, on the one hand, the efficient estimator (in Treatments T and E, where the sample average is not efficient), also based on all samples since the beginning of the session, and on the other hand, the midpoint of the range of the sample in the previous epoch; and $OR_{i,t}$ as the orthogonalized version of the latter (residuals obtained by regressing $DE_{i,t}$ onto $DA_{i,t}$). More information on these variables can be found in the supplementary information (SI) Table A.2.

	G	T	E
(Intercept)	-0.0592^{***}	$+0.0153$	-0.0166
	(0.0188)	(0.0327)	(0.0271)
$DA_{i,t}$	$+0.724^{***}$	$+0.835^{***}$	$+0.711^{***}$
	(0.0313)	(0.0239)	(0.0342)
$OR_{i,t}$		$+0.267$	-0.251^{*}
		(0.166)	(0.102)
$DE_{i,t-1}$			-0.0567^{***}
			(0.0151)
$DE_{i,t-2}$			-0.0186
			(0.0104)

	G	T	E
N_{obs}	1760	1760	1584
$adj - R^2$	0.72	0.95	0.89
BIC	452.14	2816.56	99.34
AIC	419.30	2761.83	-13.38

	G $DA_{i,t}$	T $OR_{i,t}$	E $OR_{i,t}$
SD	0.26	1.21	0.68
Lower	0.22	0.95	0.51
Upper	0.32	1.54	0.90

Table 5.2. Results of the mixed-effects model explaining deviations of submitted estimates from midpoint estimate as a function of the independent variables in the left-most column in the left panel. **Left Panel:** fixed-effects coefficient with standard errors. $^{*}p < 0.05$; $^{**}p < 0.01$; $^{***}p < 0.001$. **Right Top Panel:** number of observations, adjusted R^2, Bayesian Information Criteria (BIC) and Akaike Information Criterion (AIC). **Right Bottom Panel:** estimated standard deviation of random-effects coefficients with 95% confidence intervals (Upper, Lower limits). **G**: Gaussian; **T**: Student-t; **E**: Exponential. See supplementary information Table A.3 for the full results of fixed-effects coefficient and Table A.4 for the full results of random-effects correlation.

5.3.1 *Participant estimates revealed pervasive and significant movement away from the midpoint estimator and towards the sample average*

In all Treatments, we found that the sample average estimation was a significant predictor for the difference between participants' estimates and the midpoint; see Table 5.2; fixed-effects coefficient: Treatment G: $\beta = +0.724; p < 10^{-4}$, Treatment T: $\beta = +0.835; p < 10^{-4}$, Treatment E: $\beta = +0.711; p < 10^{-4}$. This means that participants did not simply choose the midpoint of the slider as their estimate, but instead, moved away from it in the direction of the arithmetic average based on the cumulative sample of all epochs since the beginning of a session. In Treatment G, this means that their estimates moved in the direction of highest efficiency.

Since they were all well below 1, the magnitude of the fixed-effects coefficients suggests that the average participant did not entirely eliminate the influence of the

range midpoint estimator. Even in Treatment G, with a β coefficient equal to 0.724, the average participant submitted mean estimates that lied only $\approx 3/4$ towards the sample average in the interval between the midpoint and the sample average. Curiously, in Treatments T and E, the positions of the estimates between the midpoint and sample average were not much different (βs equalled 0.835 and 0.711, respectively). In those Treatments, the sample average is not efficient, however. We report later whether any remaining difference between the actual estimate and the midpoint could be explained by efficiency.

Substantial heterogeneity appeared, suggested by the significant variability of the random coefficients, and confirmed by the better Akaike and Bayesian Information Criteria of regression models with random effects as opposed to fixed effects (see SI Table A.3 and A.4 for more information). In all Treatments, the standard deviation of the random effects was relatively low (G: SD = 0.26; T: SD = 0.21; E: SD = 0.30) when compared to the mean effects estimate reported before (see βs). Still, a few participants generated random effects that would categorize their estimation as purely following midpoints. In the G Treatment, where the sample average is an efficient estimator, these participants therefore failed to increase efficiency beyond that of the range midpoint estimator.

Fig 5.3A plots the individual random-effects estimates of the coefficient to $DA_{i,t}$ in the first ("Part 1") and second ("Part 2") replications of G. An outcome above (below) zero means that the corresponding participant is closer to (farther from) efficient estimation than the average participant. E.g., a participant with estimate equal to 0.2 has a loading equal to 0.725 + 0.200 = 0.925, and hence is close to fully efficient (which would require a loading equal to 1.0). An outcome on the vertical (horizontal) straight line implies that the participant submitted fully efficient estimates in the first (second) replication. As evident from Fig 5.3A, participants who are more efficient in Part 1 are also more efficient in Part 2 (slope of LS fitted line: 0.70; $p < 10^{-4}$). Interestingly, closer inspection of locations relative to the vertical and horizontal straight lines reveals that less efficient participants become more efficient in the second replication (Part 2), while initially more efficient participants (random effects in Part 1 above 0) remain equally efficient in the second replication. (This also explains why the estimated slope coefficient, at 0.70, is below 1.)

To verify that efficiency paid, i.e., that our experiment was well incentivized, Fig 5.3B plots the relation between the average size of the error in participants' reported predictions (|PE|) against the individual random-effects estimates of the efficiency coefficient. Prediction errors are reduced significantly as estimation efficiency increases (slope = -0.27, $p < 0.01$).

Because of the substantial heterogeneity, we performed a cohort split in terms of the random-effects estimates of the efficiency coefficient, putting into the "Efficient" group those participants whose random-effects estimates were positive in *both* replications (Parts 1 and 2), while putting all others in the "Non-Efficient" group. For Treatment G, this puts 21 participants in the "Efficient" category, while 23 were classified as "Non-Efficient." Fig 5.3C displays boxplots of the average prediction errors (|PE|) of the two groups in the two replications. The differences of the median prediction errors are significant (Wilcoxon rank sum test; p values above boxplots), and even the inter-quartile ranges do not overlap.

To gauge confidence, we studied times it took to decide whether to gamble on the true mean of the learned reward distribution or instead to choose the alternative option, a certain, known reward. Boxplots of these reaction times (in logarithm seconds) during choice are displayed in Fig 5.3D. In both replications (Parts 1 and 2), median reaction times were significantly shorter for members of the Efficient group (p values indicated above boxplots, based on Wilcoxon rank sum test). As such, efficient participants exhibited more confidence in their choices than inefficient ones.

5.3.2 *In Treatment T, estimates revealed significant movement away from the midpoint and sample average, towards the efficient estimator*

Under Treatment T, the mixed-effects regression of deviations of submitted estimates from the sample midpoint estimator (Table 5.2) revealed not only movement towards the sample average, but the remainder of the deviation could be explained for a large part (fixed coefficient = 0.267) by the direction in which the efficient estimator pointed after taking into account the direction of the sample average. That is, on average the part of participants' estimates that could not be explained by the

sample average could be attributed to variability of the efficient estimator that the sample average could not explain. Using a critical value for p equal to 0.05, however, this fixed effect was not significant. Substantial heterogeneity caused the lack of significance: the standard deviation of the random effects was estimated to be 1.21 (95% confidence interval $[0.95, 1.54]$).

Our efficient estimator derived from maximum likelihood analysis. It boils down to a simple average of a re-weighted sample, whereby the weights effectively truncate the sample depending on the estimated variability, putting less to even zero weight on outliers. The weights depend on sample variability (See Methods). Our estimator computes variability only once, however, rather than in an iterative way. Iteration based on the EM principle is usually prescribed (Liu & Rubin, 1995), but such iteration does not improve efficiency in large samples. We refrained from implementing iteration, arguing that, for humans, this would be too demanding cognitively anyway.

As with Treatment G, participants whose estimates were closer to efficient than the average (meaning that they generated a random effect above 0) tended to be more efficient in both replications of Treatment T. See Fig 5.4A: the slope coefficient in the regression of random effects in Part 2 onto those in Part 1 was a positive 0.31, with $p = 0.03$. In this Treatment too, there was evidence of improvement in efficiency upon replication. However, it concerned mostly those participants with random coefficients above 1, who over-reacted to changes in the (orthogonalized) efficient estimate; the estimates of their random effects in the replication (Part 2; vertical axis) tended to be closer to 1.

Increased efficiency reduced the size of prediction errors (Fig 5.4B) but, because of the inevitable outliers in the T environment, we reached significance only at $p = 0.07$. Median absolute predictions errors were larger for the Inefficient Group, but again, we failed to reach significance (at $p = 0.10$). Boxplots shown in Fig 5.4C show, however, that both limits of the interquartile range were lower for the Efficient participant group. Choice reaction times were indistinguishable between groups: in Part 1, mean reaction times were 11.93 (SE: 1.87) and 13.11 (SE: 1.93) for the Non-Efficient and Efficient groups, respectively. The corresponding numbers for Part 2 were 9.53 (SE: 1.60) and 10.46 (SE: 1.93). Inspection of the boxplots of the log

reaction times (Fig 5.4D) reveal that median reaction times were not significantly different across groups either. The lack of significant differences suggests a lack of increase in confidence among participants who used efficient estimation, and may be related to the lack of significance in reduction in prediction error sizes (Fig 5.4B & C). However, it could also reflect doubt among over-reacting participants (those with random coefficients above 1 in Part 1; all of those reduced their random effects in Part 2; see Fig 5.4A).

5.3.3 *In Treatment E, estimates revealed mixed evidence of movement away from the midpoint and sample average towards the efficient estimator, and decreased confidence among more efficient participants*

In Treatment E, analysis of the mixed-effects regression of deviations of individual estimates from the midpoint estimator onto the sample average and the efficient estimator proved more complicated. Per Table 5.2, first, model selection criteria (Akaike and Bayes Information Criteria) required inclusion of several (2) lags of the deviation of the efficient estimate from the midpoint when orthogonalized relative to the sample average. The resulting fixed effects ended up marginally ($p < 0.05$) (no lag) to highly ($p < 0.001$) significant (lag 1), but their signs were opposite from expected: the average participant tended to move away from the efficient estimator. Since the efficient estimator is an increasing function of the running maximum of the samples across past epochs, this means that participants on average tended to move away from this maximum, while exacerbating this move in the subsequent sampling epoch as well.

Substantial heterogeneity emerged in Treatment E too: the standard deviation of the (contemporaneous) random effects amounted to 0.68 (95% confidence interval $[0.51, 0.90]$). This actually meant that a minority of participants generated random effects of 0.25 or higher, fully compensating for the negative fixed effects ($\beta = -0.251$); for one participant, the random effect was (slightly above) 1.25, and therefore fully efficient See Fig 5.5A. Curiously though, overall there was no relationship between the random effects in the two replications, indicating that participants with efficient estimators tended to be no more efficient (relative to the mean) and

participants with inefficient estimators tended to be no less efficient upon replication. The participant who was fully efficient in Part 1 (random effect ≈ 1.2) generated a random effetc of less then -0.3 inn Part 2, and hence, was even less efficient than the average. Still, the same number of participants as in Treatment T (namely, 15) generated positive random effects in both replications, and hence, were included in the "Efficient" group. The remaining 29 were put in the "Non-Efficient" group for Treatment E.

Participants who implemented more efficient estimation generated significantly smaller absolute prediction errors ($p < 0.01$, both replications) and hence higher earnings. See Fig 5.5B and C. Surprisingly, choice reaction times for participants in the Efficient group were significantly higher than those for the Non-Efficient group (median comparison based on Wilcoxon rank sum test, see Fig 5.5D; mean reaction times in Part 1 for Efficient and Non-Efficient participants were 13.50s and 6.97s, respectively; this difference is significant at $p < 0.01$ based on a two-sample t test; mean reaction times in Part 2 for Efficient and Non-Efficient participants were 10.27s and 5.75s, respectively; this difference is significant at $p < 0.01$ based on a two-sample t test).

5.3.4 *Choices revealed risk neutrality and generally were consistent with submitted estimates*

Assuming risk neutrality, we investigated whether choices were consistent with submitted estimates of the mean of the gaming machines with random outcomes. Participants had to choose whether to be paid the true mean of the random outcomes in a replication, or an alternative, known and fixed reward. If a risk-neutral participant estimated the (unknown but estimated) mean to be higher than the fixed reward, the former should have been chosen, and *vice versa*.

Table 5.3 lists the results when stratifying the participants into an "Efficient" and an "Non-Efficient" group. Grouping is obtained per treatment, in the manner explained before. The table lists the number of participants in each group/replication/treatment. Very few participants made choices that were inconsistent with their reported estimates, regardless of whether they belonged to the Efficient or Non-Efficient group,

	G	T	E
Efficient	21	15	15
Non-Efficient	23	29	29

	G		T		E	
Part	P1	P2	P1	P2	P1	P2
Efficient	0	1	0	0	1	1
Non-Efficient	0	0	7	2	0	1

Table 5.3. Left: number of participants in Efficient and Non-Efficient groups, per Treatment. The Efficient group includes only participants who generated positive random coefficients for the (orthogonalized) efficient estimation regressor in Table 5.2. **Right:** Per replication, number of inconsistent choices, i.e., choices that, under risk neutrality, contradict an individual's final reported estimate of the expected reward for the risky option relative to the reward offered on the risk-free option. Treatments: **G**: Gaussian; **T**: Student-t; **E**: Exponential.

with one exception: 7 (out of 29) participants in the Non-Efficient group made choices in Part 1 of Treatment T that were inconsistent with their estimates. These inconsistencies disappeared in the second replication (Part 2), however.

To explain the discrepancy for the Non-Efficient group in the first replication of Treatment T, we studied to what extent this group encountered more outliers (which could be substantial; see example in Fig 5.1A) than the Efficient group. We defined an outlier as having occurred if the sampled outcome was located relative to the mean at more than 1.5 times the standard deviation. There was, however, no difference in the number of outliers (see SI Table A.5).

5.4 DISCUSSION

We discovered how the average participant moved away from an estimator that required the least mental and motor effort, namely, the midpoint of the range of outcomes, and towards more efficient estimators, in the first place the sample average, and in our Treatment T, towards a properly re-weighted sample average. Treatment T presented a situation where the "Pareto Principle" held: outliers were large and frequent, deteriorating the efficiency of both the midpoint estimator and the (unweighted) sample average.

We also documented substantial heterogeneity: some of our participants effectively stayed with the midpoint estimator, while others submitted fully efficient estimators. In the Gaussian (G) treatment, efficiency led to significantly higher per-

formance; in the T Treatment, even if prediction errors were lower for participants who submitted more efficient estimates, the difference was not significant because of the frequent outliers. Moreover, some participants over-reacted to the efficient estimator in the first replication (but corrected this in the second replication). Evidently, these two facts affected confidence: while in G efficient participants chose with more confidence (the time it took them to decide whether to gamble on the magnitude of the true mean rather than receiving an exogenous, fixed reward), no such difference in confidence was observed in the T Treatment.

Participants appeared to struggle with the proper way of obtaining statistical efficiency in the E Treatment. There, outcomes were drawn from a negatively skewed distribution. Because the distribution, the exponential distribution, featured an (unknown) maximum that changed from one replication to the next one, the mean changes with the maximum. Extremum estimators converge very fast (at a rate proportional to the size of the sample), unlike the sample average (which converges at a rate proportional to the square-root of the sample size). Consequently, the most efficient estimator tracks the in-sample maximum. The average participant appeared to sense indeed that there was a relation with the maximum: their estimates correlated with the running maximum. But some participants tended to react in the opposite direction to efficient estimation, lowering the estimate when the maximum increased and *vice versa*. Only a few participants managed to submit fully efficient estimates. When they did so, they mostly reverted to inefficient estimation in the second replication. As such, the evidence points to confusion. This then translated into lower confidence (longer choice reaction times) among the Efficient group of participants.

Note however that, in Treatment E, 1/3 of the participants could be classified as "more efficient than the average" in both replications (Parts 1 and 2). This proportion was not lower than in Treatment T, and only slightly lower than in Treatment G.

Human and non-human animal learning of stochastic rewards has been studied extensively with mathematical models of reinforcement. The models have proven successful, not only in shedding light on animal adaptation, but also in allowing for fruitful interrogation of neural signals in the human brain, especially in the

dopaminergic system (Schultz et al., 1997; Dabney et al., 2020). The models have provided a basis for effective artificial intelligence (Sutton & Barto, 2018).

These models invariably assume that expectations are formed recursively, or, more recently, through integration over a fitted distribution (Dabney, Rowland, et al., 2018). Our study shows that human learning is not exclusively based on sample averages as estimates of expected rewards, but that concern for statistical efficiency (in the sense of Cramér and Rao (Cramér, 2016) and Chapman and Robbins (Chapman et al., 1951)) leads humans to search for, and implement, better estimation of the expectation. This may prove fruitful for Artificial Intelligence: novel reinforcement learning protocol could be developed that works better than traditional, sample-average-based procedures when more efficient estimators (of the expectation) exist. This is the case in finance, for instance, were most reward (return) distributions satisfy the "Pareto principle," and hence, are heavy-tailed. There, an extension of distributional reinforcement learning where expectations are formed based on maximum-likelihood estimation has been shown to be more robust to tail events (Bossaerts, Huang, & Yadav, 2020).

There exists an obvious link between our study and the literature on model-based reinforcement learning, where the agent posits that observable values are driven by a hidden model that, if used in inference, vastly improves learning speed (Daw, Gershman, Seymour, Dayan, & Dolan, 2011). It has yet to be determined, however, what overarching principle governs the search for an appropriate model. Recent studies have focused on instrumental aspects of a task (Botvinick & Weinstein, 2014). Much work has to be done, however, on the purely predictive aspects. In the context of medical assessment, these predictive aspects concern the hidden cause of a symptom such as fever. If a doctor knows the hidden cause of fever, she can predict more accurately the effect of antibiotic treatment. Episodic memory obviously plays a crucial role; episodic memory helps identifying the right model on which to base reinforcement learning (Botvinick et al., 2019).

Episodic memory may also have played a role in causing the heterogeneity we recorded. Participants who have been exposed regularly to negative skewness may understand better how to exploit extremum statistics such as the sample maximum to improve the efficiency of estimation of the mean. In the future, it may be necessary

to query participants on familiarity with the types of risks we expose them to in the laboratory. Differential costs to cognitive effort may be another reason why heterogeneity emerged in our study. In our paradigm, the range midpoint is the cheapest to compute and implement. Not surprisingly, some participants therefore chose to follow this estimator, deciding to not even attempt prediction based on the sample average.

We conjecture, boldly, that the search for a model of the environment is driven by an overarching concern for statistical efficiency. Model building and improvement is expected to occur when efficiency gains are possible, i.e., when more information can be extracted from available samples. In formal terms, this means that human reinforcement learning is governed by a desire to reach lower bounds to standard errors, such as the Cramér-Rao or Chapman-Robbins bounds. As the environment changes, lower bounds change, and learning performance should change as well, inducing participants to extract information from available samples in different ways. The search for improved statistical efficiency is not straightforward however, as our results for Treatment E showed. We expect it to be a dynamic process, whereby the agent looks at several aspects of a sample, including range midpoint, sample average and order statistics, building confidence only when sensing improved efficiency, while losing confidence when failing to discover the right way to improve efficiency. Evidence has emerged recently that reinforcement learning is a dynamic process whereby humans search for the contextually best model (da Silva & Hare, 2020).

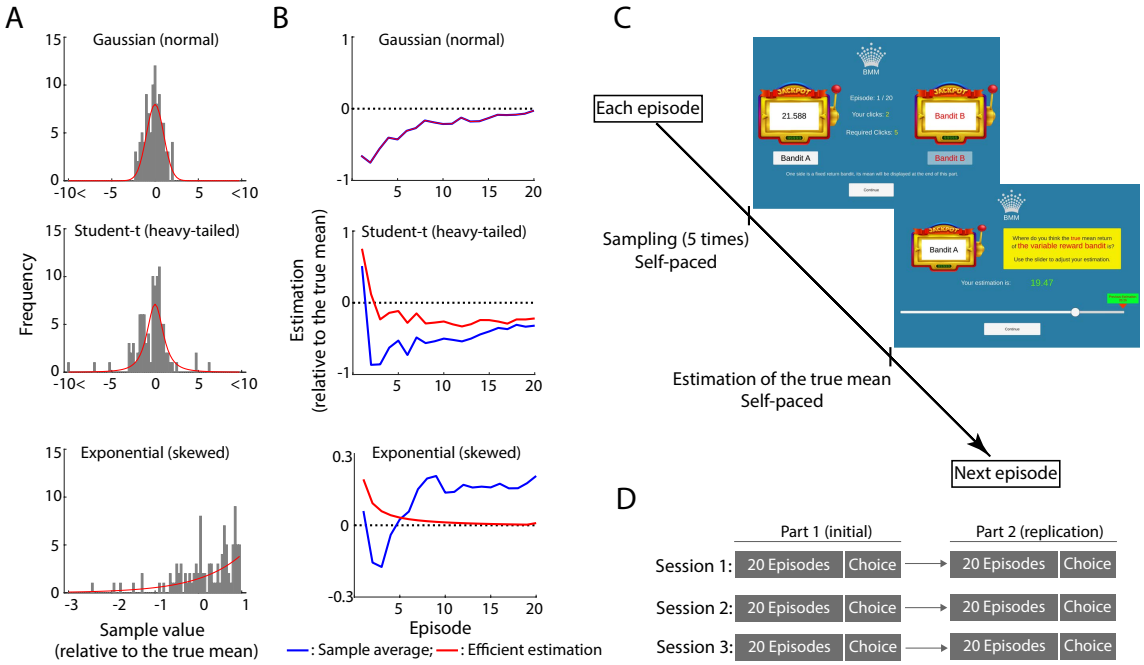


Figure 5.1. **A.** Histogram of de-meaned draws from an example participant/replication. **B.** Time series of de-meaned estimations (estimation - true mean) by the sample average method (blue) and the efficient estimation method (red) based on all sampled values displayed to an example participant/replication up to episode i. Note in the Gaussian Treatment (the top graph), the two lines overlap because the sample average estimator is fully efficient. **C & D.** Timeline of the experiment. The experiment consisted of three sessions. In each session, one distribution was randomly selected without repetition to provide the draws of the risky gaming machine. Each session contained two parts (replications). Each part contained twenty episodes of sampling and estimation tasks, plus one choice task. In the sampling task, participants were required to click a button to sample five times. Subsequently in the estimation task, participants were asked to use the slider to provide their estimation of the true mean of the gaming machine. Participants repeated the two tasks for twenty episodes and then made a choice between a fixed risk-free value and the risky gaming machine. In the repetition part, the nature of the gaming machine remained the same while the true mean changed. The positions of the risky and the risk-free gaming machine swapped. See Methods for details.

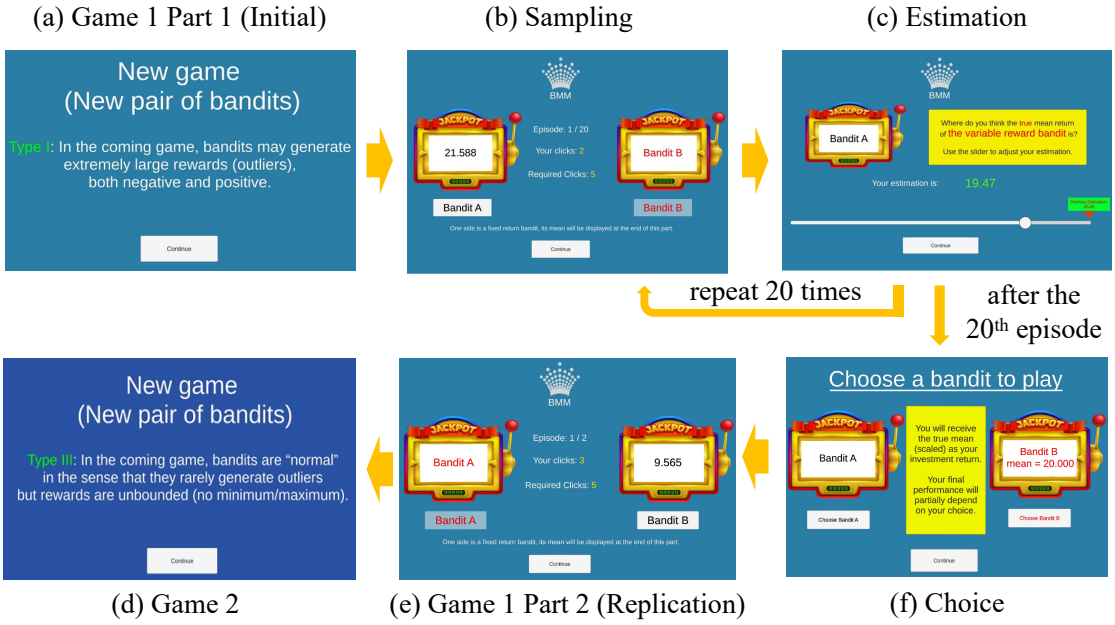


Figure 5.2. Experiment design. (a) & (d) Participants started a new game with some information about the nature of the coming values and a change of the background color. See supplementary information Table A.9 for true mean values and Table A.10 for distribution and background color assignments. (b) Sampling task: participants clicked only one button (left button in the illustration) to sample the values. The other button is disabled (greyed out). (c) Estimation task: participants used the slider to indicate where they thought the true mean was. From the second ($t = 2$) episode, participants' previous estimation ($t - 1$) was displayed in a green box on top of the current slider. The green box was located at where the previous ($t - 1$) estimation would be on the current (t) slider. (f) Choice task: participants were to choose between the risky and the risk-free gaming machine. The mean of the risk-free gaming machine was displayed to the participants at this point. (e) Repeat: in part 2, the position of the gaming machines swapped; the mean values were changed but the nature of the distribution remained the same.

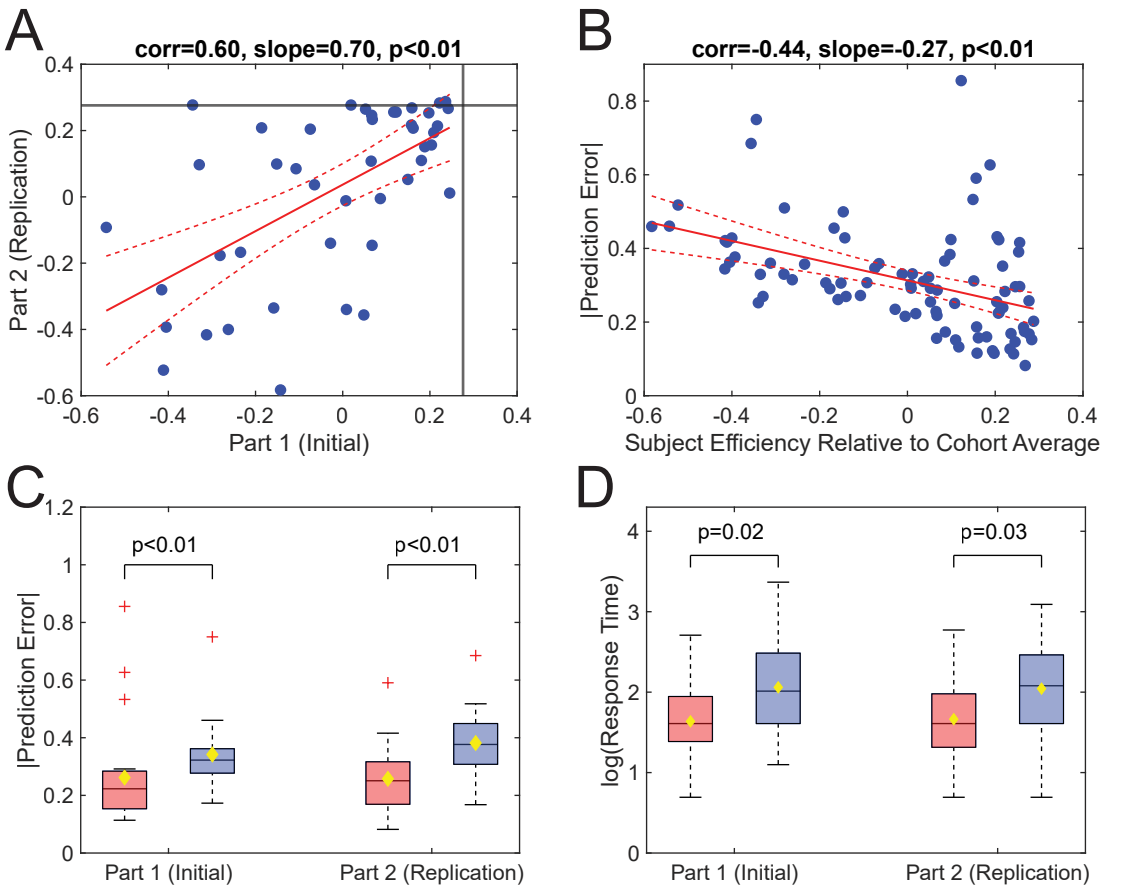


Figure 5.3. Gaussian Treatment (G). A. Relationship between individual deviations from cohort mean in impact of sample average estimation on estimates submitted by the participants in first (Part 1) and second (Part 2) replications (i.e. random effects coefficients for the regressor $DA_{i,t}$, the difference between the sample average estimation and the midpoint of the slider). The horizontal and vertical straight lines show the theoretical most efficient case in each replication. Values to the right of (below) the straight line imply less than fully efficient estimates in Part 1 (Part 2). **B.** Relationship between individual deviations from cohort mean in impact of sample average estimation on submitted estimates and average size of prediction errors $|\text{PE}_{i,t}|$ in both replications. **A & B.** Red straight lines represent fitted lines. Red dashed lines correspond to the upper and lower confidence bounds at the 95% level. In figure titles we report the correlation between the data of the X-axis and the Y-axis (Corr), the slope of the fitted line and its corresponding p value. **C & D.** Boxplots. Red boxes are for the efficient group (participants with positive random coefficients in both replications, as per figure in Panel A). Blue boxes are for the non-efficient group. Yellow diamonds represent the mean of the data. The p-values were obtained from the Wilcoxon rank sum test between the two groups. Red crosses represent outliers that are more than 1.5 times larger than the limits of the inter-quartile range.

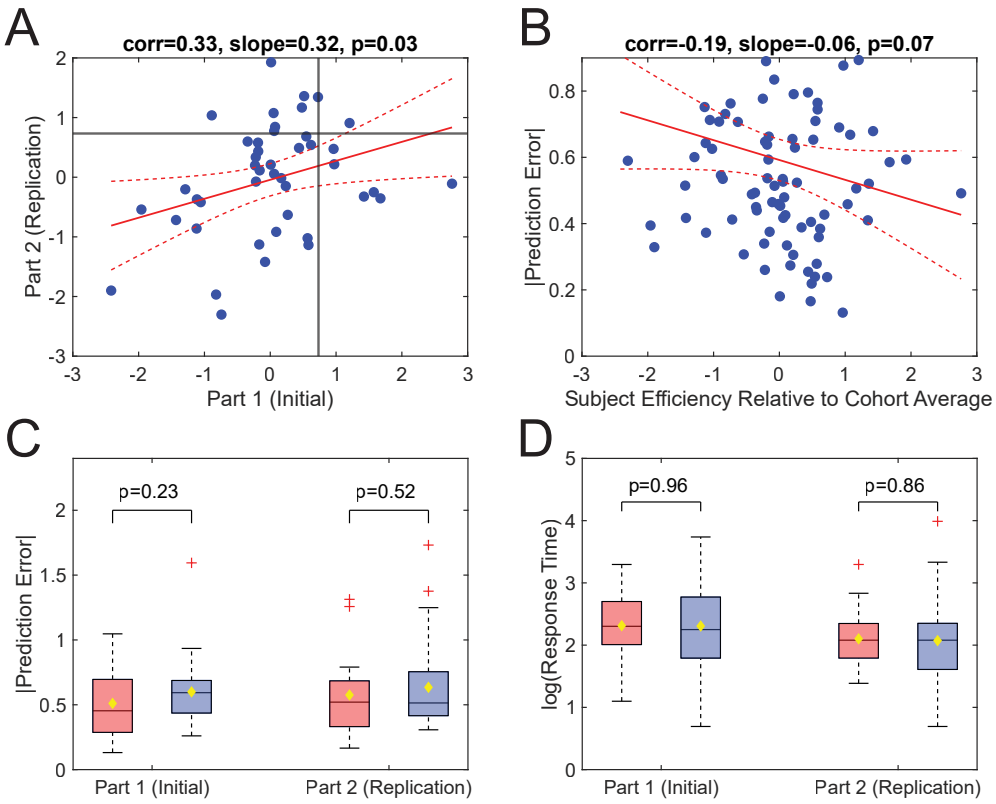


Figure 5.4. Student-t Treatment (T). A. Relationship between individual deviations from cohort mean in impact of efficient estimation on estimates submitted by the participants in first (Part 1) and second (Part 2) replications (i.e. random effects coefficients for the regressor $OR_{i,t}$, the orthogonal residuals obtained from regressing the difference between the efficient estimation and the midpoint of the slider, $DE_{i,t}$, on the difference between the sample average and the midpoint, $DA_{i,t}$). The horizontal and vertical straight lines show the theoretical most efficient case in each part. Values to the left of (below) the straight line indicate less than efficient estimation for Part 1 (2). **B.** Relationship between individual deviations from cohort mean in impact of efficient estimation on submitted estimates and average size of prediction errors $|\text{PE}_{i,t}|$ in both replications. **A & B.** Red straight lines are fitted lines. Red dashed lines correspond to the upper and lower confidence bound at the 95% level. In figure titles, we report the correlation between the data on the X-axis and Y-axis (Corr), the slope of the fitted line and the corresponding p value. **C & D.** Boxplots. Red boxes are for the efficient group (participants with positive random coefficients in both replications). Blue boxes are for the non-efficient group. Yellow diamonds represent the mean of the data. The p-values were obtained from the Wilcoxon rank sum test between the two groups. Red crosses represent outliers that are more than 1.5 times larger than the limits of the inter-quartile range.

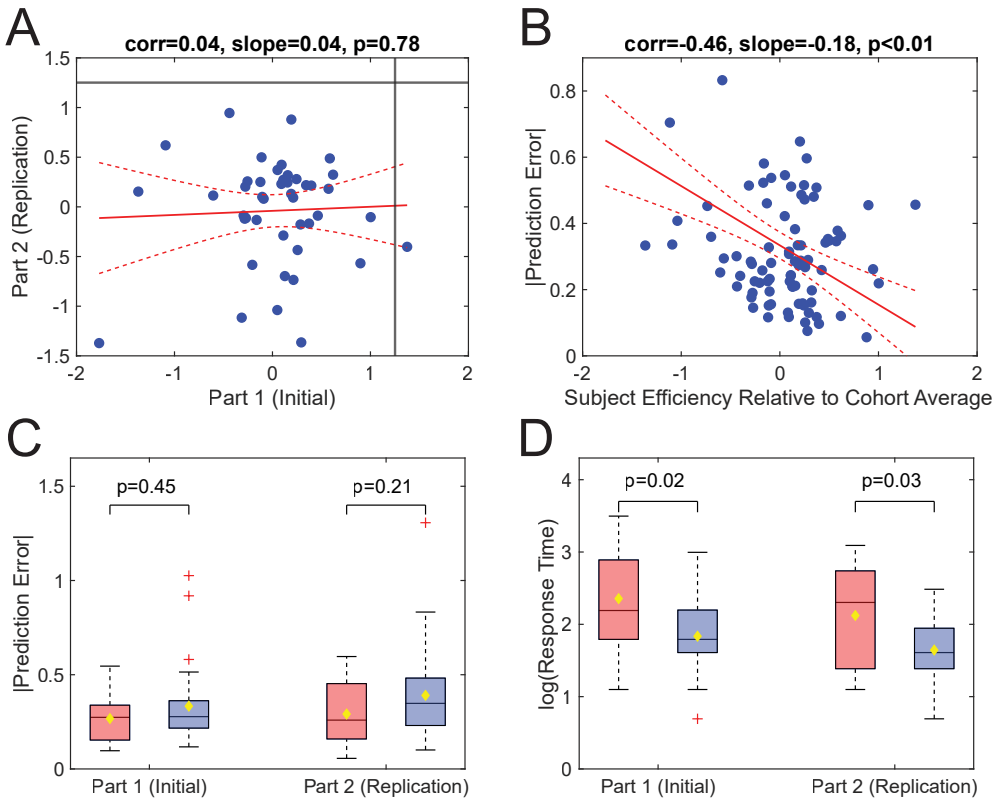


Figure 5.5. Exponential Treatment (E). A. Relationship between individual deviations from cohort mean in impact of efficient estimation on submitted estimates in first (Part 1) and second (Part 2) replications (i.e. random effects coefficients for the regressor $OR_{i,t}$, the orthogonal residuals obtained from regressing the difference between the efficient estimation and the midpoint of the slider, $DE_{i,t}$, on the difference between the sample average and the midpoint, $DA_{i,t}$). The vertical (horizontal) straight lines show the position of the theoretically most efficient case in Part 1 (Part 2). Values to the left (below) the straight line imply that submitted estimates were less than fully efficient. **B.** Relationship between individual deviations from cohort mean in impact of efficient estimation on submitted estimates and average size of prediction errors $|\text{PE}_{i,t}|$ in both replications. **A & B.** Red straight lines are fitted lines. Red dashed lines correspond to the upper and lower confidence bounds at 95% level. In the title of the figures, we report the correlation between the data of the X-axis and the Y-axis (Corr), the slope of the fitted line and its corresponding p value. **C & D.** Boxplots. Red boxes are for the efficient group (participants with positive random coefficients in both replications). Blue boxes are for the non-efficient group. Yellow diamonds represent the mean of the data. The p-values were obtained from the Wilcoxon rank sum test between the two groups. Red crosses represent outliers that are more than 1.5 times larger than the limits of the inter-quartile range.

Part V

ORIGIN OF TAIL RISK

6

ORIGIN OF TAIL RISKS

6.1 INTRODUCTION

In experimental and agent-based economics, it has been central to gauge how much economy-wide information is needed at the individual level to induce Pareto efficiency and generate stylized features in financial markets (Arrow, 1951; Debreu, 1951, 1954; Smith, 1962; Cliff, 1997). Do traders who utilize more economy-wide information, such as trade price and volume, benefit the market? Does this come at the cost of increased tail risk? This paper studies these two questions using a real-time trading simulation paradigm.

The first question concerns economic welfare and how, if possible, markets equilibrate to Pareto optimum through interactions among traders, including humans and automated trading robots. The field that studies the interactions among human traders is *experimental economics*, pioneered by Smith (1962). More recently, the rise of automated trading robots and artificial intelligence has intrigued both academics and the financial industry to study interactions and market efficiency among such robots, and between humans and robots (Duffy, 2006; March, 2019; Asparouhova et al., 2020; Bao, Nekrasova, Neugebauer, & Riyanto, 2021). The field called agent-based economics focuses exclusively on robots, merging insights from economics and computer science.

The second question examines how tail risk emerges. Tail risk is usually measured in terms of leptokurtosis (although, as we shall demonstrate, the two concepts are not synonymous). Formally, a distribution is leptokurtic when its density mass

of the distribution cluster in both the center and the tails of the distribution. The distribution of price changes in financial markets is known to be leptokurtic[1]. Historically, due to financial crisis and concomitant regulations, the literature has shown a considerable amount of interest in modeling the tails using statistical and econometrics methods, e.g., value-at-risk (Sims, 1980; Rockafellar & Uryasev, 2000) and power-law (Gabaix et al., 2003; Clauset et al., 2009). However, the origin of leptokurtosis remains elusive (Farmer & Lillo, 2004). Here, we study to what extent leptokurtosis is solely the consequence of the interaction of agents through a standard microstructure, namely, the continuous open-book system.

We address the two questions using a paradigm known as *Zero-Intelligence* from Gode and Sunder (1993) due to its clean set-up and compact connection to Smith (1962). In this paradigm, all trades occur in a continuous open-book system populated with so-called Zero-Intelligent Traders (ZITs). Here "intelligence" is defined as the ability to trade on system-wide information, such as trade price and volume. ZITs are traders who place orders with no specific strategies as long as the trades are profitable via arbitrage[2]. The definition of "intelligence" connects with the literature on *swarm intelligence*, which refers to the ability of a group of decision-makers to interact with their environment in ways that generate apparent intelligence at the group level (X.-S. Yang et al., 2016). Crucially, decision-makers do not use system-wide information[3].

Gode and Sunder (1993) show that the economic welfare generated through trading interactions among ZITs is very high, but a closer inspection reveals that it is strictly inferior to that generated by human traders, except for a special case where the demand-supply curve of the underlying economy is symmetric and the agents' profit margins are strictly zero (Cliff, 1997; De Luca, Szostek, Cartlidge, & Cliff, 2011).

The result has prompted research on how much information at the individual level is needed in order to generate as much, or even more, welfare as human traders

[1]For instance, the distribution of daily price changes of S&P500 stock market index from 2010 to 2022 has a kurtosis of 13, as opposed to a Gaussian distribution, which has a kurtosis of 3.
[2]There are different types of ZITs in the agent-based literature. The type we refer to is called "ZITs-constrained" where the only constraint is to make profitable traders.
[3]"The rules that govern interactions among [participants] are executed on the basis of local information that is without knowledge of the global pattern" (Garnier et al., 2007).

while at the same time enhancing the use of system-wide information (e.g., order flow, trade prices) to benefit individual traders maximally. Recently, the focus of attention has been the latter: contests have been set up to reward the best-performing trading algorithm[4]. Here, we revert to the original economic question, and study to what extent access to system-wide information enhances welfare, if at all.

To do so in an incremental fashion, we introduce a common market participant, a market-maker without any other incentive to trade than to profit from access to system-wide information, while simultaneously attempting to remain inventory neutral. The U.S. Securities and Exchange Commission (SEC) defines a market-maker as "a firm that stands ready to buy or sell a stock at public quoted prices"[5]. Their primary business model is to earn a slight difference, called bid-ask spread, between the sell (ask) and buy (bid) price but do so with a massive volume (Weaver, 2012). By frequently quoting bid and ask prices, market-makers are said to supply liquidity to financial markets, and hence are often referred to as liquidity providers.

Traditionally, the discipline that predominately studies market-makers and market liquidity at a granular level is *market microstructure*. Broadly, market microstructure can be bifurcated into *theoretical* and *empirical* market microstructure. The former theorizes security price dynamics, agents' behavior, and trading mechanisms "when trading is impeded by frictions, such as information, decision and transaction costs" (K. J. Cohen, Maier, & Whitcomb, 1986). The types of friction concerned by theoretical microstructure are mainly limited to two: inventory constraints (e.g., Garman (1976); Amihud and Mendelson (1980); Ho and Stoll (1981)) and information efficiency (e.g., Bagehot (1971); Glosten and Milgrom (1985); Kyle (1985); Duffie, Gârleanu, and Pedersen (2005); Lagos and Rocheteau (2009)).

Empirical market microstructure uses theoretical models to explain empirical observations, e.g., price discovery and price formation dynamics. The research is dispersed, varying in topics and asset classes. Examples are factors that determine bid-ask spread dynamics (Hasbrouck & Seppi, 2001; Comerton-Forde, Hendershott,

[4]e.g., the Santa Fe contest (Palmer, Arthur, Holland, LeBaron, & Tayler, 1994) and most recently, Jane Street Market Prediction (https://www.kaggle.com/c/jane-street-market-prediction) and Optiver Realized Volatility Prediction (https://www.kaggle.com/c/optiver-realized-volatility-prediction).

[5]https://www.sec.gov/fast-answers/answersmarkethtm.html

Jones, Moulton, & Seasholes, 2010), policy implications on market making (Anand, Tanggaard, & Weaver, 2009; Bessembinder, Panayides, & Venkataraman, 2009) and the impact of high-frequency liquidity provider on market liquidity (Hendershott, Jones, & Menkveld, 2011; Brogaard, Hendershott, & Riordan, 2014).

Despite a bountiful market microstructure research, there exists a lacuna between the two subfields. Theoretical market microstructure relies on unobservable factors (such as private information), to which the empiricists have no access. In practice, empiricists have leaned upon proxies to circumvent issues of unobservable factors. In addition, most theories are stylized, ignoring the flexibility and hence, huge strategy space, afforded by the continuous open-book system. The advent of computerized trading has made the gap between theory and empirical studies even larger due to emerging ambient issues such as order submission latency and concurrency[6].

Here, we demonstrate that modeling of the kind encountered in experimental economics and agent-based economic analysis can be used to shed light on issues that occupy market microstructure empiricists. In our paradigm, a simulated economy is populated with ZITs who react only to private demand-supply incentives and do not utilize economy-wide information, such as trade price and volume data. The fundamentals of the economy are Gaussian. Specifically, demand and supply shifts occur regularly, and are such that equilibrium price changes are Gaussian. As such, tail risk and leptokurtosis can occur because agents trade at off-equilibrium prices. But such off-equilibrium trades may also lower allocation (Pareto) efficiency. The modeling builds on the examples in Alton and Plott (2007) and Cvitanić, Plott, and Tseng (2015). We then introduce a liquidity provider. The liquidity provider is the only agent who has access to system-wide information: it seeks to profit from the bid-ask spread by learning economy-wide information yet remains inventory neutral. All participants trade in a continuous open-book system with the presence of order submission latency and concurrent actions.

We find that while the liquidity provider tightens the bid-ask spread and facilitates trades, it does not uniformly improve economic welfare, measured by Pareto efficiency. Moreover, we demonstrate that in the absence of a liquidity provider, trading generates leptokurtosis, and hence excessive tail risks. Introduction of the

[6]For instance, Menkveld and Zoican (2017).

profit-seeking liquidity provider increases leptokurtosis further, but tail risks are not worsened.

While both Aldrich and López Vargas (2020) and Asparouhova et al. (2020) have market making algorithms in place in an experimental setting, this paper differs from them twofold: (1) human subjects chose whether to use market making algorithm, whereas here the decision-maker is purely algorithmic, and (2) while market-makers are also bounded by budget constraints, they do not actively manage inventories. In this paper, in addition to budget constraints, active inventory risk management is crucial and inherent in the market making algorithm.

The rest of this paper is organized as follows. Simulation set-up is presented in Section 6.2. Analysis methodology is in Section 6.3. Results are reported in Section 6.4. Discussions and empirical implications are discussed in Section 6.5.

6.2 SIMULATION DESIGN

In this section, we describe the simulation design. We introduce the underlying economy and markets, then elaborate on agents' design, and finally lay out the procedures.

6.2.1 *The economy: incentive and pricing mechanism*

We consider a single-widget economy that comprises a group of agents, including twenty zero-intelligent traders (ZITs), a demand-supply inducer (DS inducer), and a market-maker (MM). The role of the DS inducer is to construct the economy by providing economic surplus to ZITs in order to incentivize trading activities. ZITs are purely arbitrage traders; they receive private signals from the DS inducer at the beginning of each period $m \in [0..M]$, and then trade widgets for arbitrage profits. The market-maker is an intelligent trader who utilizes only public information such as trade prices and volume to make profits. We refer readers to Subsection 6.2.3 for detailed design of agents.

To model private signals and the shift of equilibrium, we follow examples in Alton and Plott (2007) and Cvitanić et al. (2015) and implement stepwise demand and

supply function (see Figure 6.1A for an example). Formally, each demand signal (bid price) $D_{i,m}$, $i \in [0..10]$ and each supply signal (ask price) $S_{j,m}$, $j \in [0..10]$, $m \in [0..M]$ are modeled as the following:

$$D_{i,m} = D^h_m - L_D * i \tag{6.1}$$

$$S_{j,m} = S^h_m + L_S * j \tag{6.2}$$

where D^h_m (S^h_m) is the highest private demand (supply) incentive at m[7]; D^h_m and S^h_m are considered as the "reference price". L_D and L_S are the step size; we set $L_D = L_S = 50$ (cents). When the DS inducer sends a bid (ask) as a private signal, it provides the highest (lowest) possible bid (ask) that the ZITs are willing to trade in the public market before incurring losses. Notice that under this stepwise demand and supply setting, the equilibrium price P^* can be a range; the equilibrium quantities are a scalar, namely the minimum of the demand quantity and the supply quantity at the equilibrium price.

Across different periods, we shift the highest bid and ask price:

$$D^h_{i,m+1} = D^h_{i,m} + \Delta D^h \tag{6.3}$$

$$S^h_{j,m+1} = S^h_{j,m} + \Delta S^h \tag{6.4}$$

where ΔD^h and ΔS^h are randomly drawn from a Gaussian distribution[8]:

$$\Delta D^h \sim N(0, \frac{L_D}{2}) \tag{6.5}$$

$$\Delta S^h \sim N(0, \frac{L_S}{2}) \tag{6.6}$$

The equilibrium may or may not shift as a result of the demand-supply function shift. All private signal prices are rounded to the nearest integer (cents). An illustration

[7] or the slope if both the demand and the supply functions are linear.

[8] Strictly speaking, the ask price must be manually bounded below to ensure that all prices are greater than zero. We achieved this by imposing a maximum gap between the highest bid and ask price.

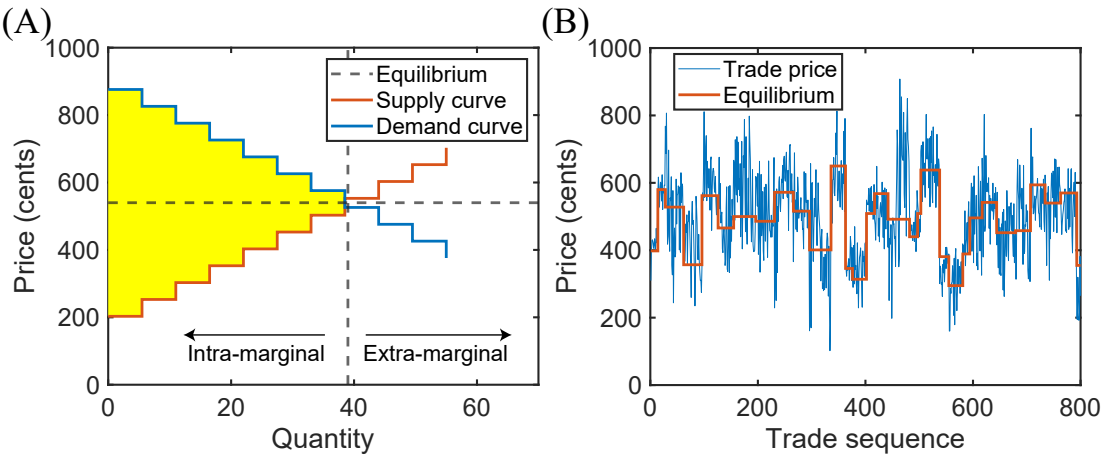


Figure 6.1. Economy and regime shift. (A) The plot illustrates the underlying economy in a single period. The orange step function is the supply curve. The blue step function is the demand curve. The equilibrium quantity is 39. The demand equilibrium price is 576 cents whereas the supply equilibrium price is 503 cents. Each step represents a private signal to a zero-intelligent trader. The yellow shaded area is the theoretical maximum total economic surplus in that period. The economy is said to reach Pareto optimal state when the aggregate actual surplus obtained from trading activities equals the theoretical maximum surplus. See Subsection 6.2.1 for a detailed documentation of how agents' incentives and the pricing mechanisms are determined. **(B)** The plot shows the price series of the first 800 public trades (blue time series) and the equilibrium price series when each trade occurred in a sample ZITs-only session (orange step curve).

of a single-period demand-supply function, as well as economy regime shifts, is given in Figure 6.1.

6.2.2 *Market mechanism - a continuous open-book system*

All trades take place in a continuous open-book system, a generalized double auction in which orders from intra-marginal traders remain in a book until executed or canceled, whichever comes first. There are two types of markets, *private* and *public*. One security called "widget" is traded in both markets. Widgets can be bought and sold on a public *limit order book* (LOB) available to all traders in the public market.

The LOB has two sides, *bid* and *ask*[9]. The bid side records all outstanding buy orders, while the ask side records all outstanding sell orders. The public market is anonymous; trader identification is available only for analysis.

The *best bid* refers to the best price at which someone is willing to buy a widget, namely the **highest** buy price. Similarly, the *best ask* refers to the best price at which someone is willing to sell a widget, namely the **lowest** sell price. The term *bid-ask spread*, or in short the *spread*, equals the best ask minus the best bid; as a result, the spread is always greater than zero by definition. A trade occurs if the price of a new order crosses the best price on the opposite side, i.e., when the price of the incoming bid is higher than or equal to the current best ask or when the price of the incoming ask is lower than or equal to the current best bid. Otherwise, the incoming order joins the queue of the corresponding side in the LOB. The commission fee is assumed to be zero in these simulations. All outstanding orders, or submitted but non-traded orders, are listed in the LOB and sorted in descending order. See Figure A.3 in the appendix for the illustration of an example LOB.

In the meantime, each ZIT has its designated private market to receive private signals from the DS inducer. A private signal is either a private bid or a private ask order. This paper uses the terms private signal and private order interchangeably. The private market setting mirrors the public market except for who has access to the information. All traders have access to the order book in the public market anonymously[10], yet only the DS inducer and the corresponding ZIT are privy to the information in each private market.

A schematic diagram that summarizes the economy and market structure is given in Figure 6.2.

6.2.3 *Algorithmic traders*

We introduced three types of algorithmic traders in this economy: a DS inducer ($N = 1$), twenty zero-intelligence traders (ZITs) ($N = 20$), and a market-maker

[9]Also known as "bid and offer".

[10]Technically, ZITs have access to the public market information, but by design, they choose to ignore all public information.

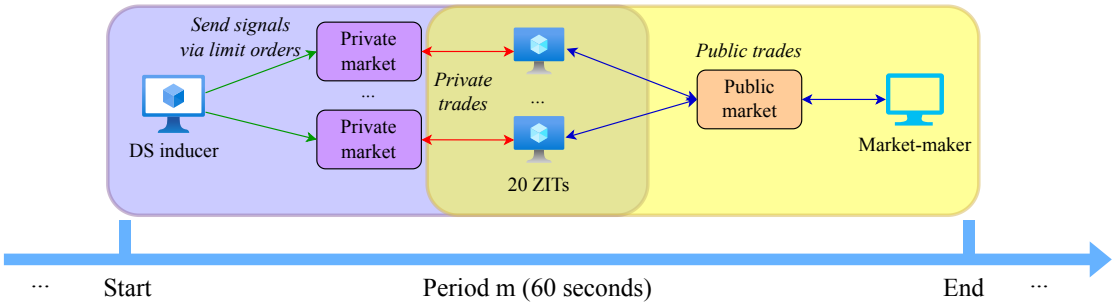


Figure 6.2. Schematic diagram of the simulation. There are two types of markets, *public* and *private*. The demand-supply inducer (DS inducer) interacts with each ZIT $i \in [0..20]$ via an exclusive private market; private signals are not visible to other ZITs and the market-maker. After receiving a private signal, each ZIT submits a public bid/ask order in the public market. Pseudocode of ZITs is given in Algorithm 6.1. An example with hypothetical numbers is given in Figure 6.3. The market-maker can only observe the public market and submits bid-ask order pairs as per Algorithm 6.2.

(MM) ($N = 1$). Below we lay out the details of each type. A tabular summary can be viewed in Table 6.1.

Zero-intelligence traders (ZITs): the sole mission of ZITs is to earn profit from arbitrage, that is, by buying low in one market and selling high in the other market. At the beginning of each period m, each ZIT will receive its private signal $P_{ZIT}^{private}$ of five widgets via its private market; it then proceeds to the public market to buy (sell) widgets for a lower (higher) price. Each ZIT submits only one widget per public order at a time and waits for the outstanding order to be traded before sending another one. The public order price, P_{ZIT}^{bid} or P_{ZIT}^{ask}, is determined randomly by a scaled beta distribution $Beta(\alpha, \beta)$ between its private signal and the minimum market price (or maximum if the private signal is on the ask side). We set $\alpha = 1.5$, $\beta = 10$; the minimum order price is 0 (cent) and the maximum order price is 1000 (cents).

	Zero-Intelligent Agent	**Liquidity Provider**
No. of agents	$N = 20$	$N = 1$
Description	1. Know individual private signal via a limit order with five units from the DS inducer. 2. Ignore system-wide information.	1. Has to calculate the widget's fair value from the public market. 2. Uses system-wide information.
Market access	Public and private.	Public only.
State input	Private signal only.	System-wide information.
Action	1. Place public bid/ask order. 2. One widget per order. Wait for an order to be consumed before sending another.	1. Places a bid-ask order pair. 2. Places inventory rebalance order. 3. One widget per order.
P&L	1. Profit: arbitrage, buy low from one market and sell high in the other. 2. Loss: none	1. Profit: bid-ask spread. 2. Loss: inventory rebalance.

Table 6.1. Zero-intelligent traders (ZITs) and the liquidity provider. The state input indicates the information the robots can observe and use. The action describes the robots' actions, conditional on state input. The P&L describes the profit and loss incurred as a result of the actions performed. Note that the DS inducer provides the profit (surplus) for all the trader robots in this economy. Hence, it is expected to make a loss. System-wide information consists of all information available in the market, such as the order book, all past trades, and the trade volume. Technically speaking, ZITs can observe state wide information. However, they do not use state wide information.

$$P_{ZIT}^{bid} \sim Beta(x, \alpha, \beta) \text{ where } x = \frac{P_{ZIT}^{bid}}{P_{ZIT}^{private}} \tag{6.7}$$

$$P_{ZIT}^{ask} \sim Beta(x, \alpha, \beta) \text{ where } x = \frac{P_{ZIT}^{ask} - P_{ZIT}^{private}}{1000 - P_{ZIT}^{private}} \tag{6.8}$$

If the public order is traded, the ZIT immediately trades the same widget with the DS inducer in its designated private market. The dollar difference between the two trades is captured by the ZIT as the arbitrage profit. If all five widgets in the private signal are consumed, the ZIT will stay inactive for the remaining time of the period. Otherwise, all outstanding public orders from ZITs will be canceled at the end of each period.

We provide a graphical description of a typical ZIT robot in Figure 6.3. See Algorithm 6.1 for the pseudocode of ZITs.

Algorithm 6.1: Zero-Intelligent Trader (ZITs)

```
1  Function Main():
2      Receive private signal from the DS inducer: a private order of 5 units;
       /* initialisation of parameters                                    */
3      units := 0;
4      while units < 5 do
5          if no pending public order then
6              if private signal is demand then
7                  bid_price := randomly draw from a beta distribution between the
                     private signal and minimum order price (0 cent);
8                  submit a bid order for 1 unit at bid_price;
9              else if private signal is supply then
10                 ask_price := randomly draw from a beta distribution between the
                     private signal and maximum order price (1000 cents);
11                 submit an ask order for 1 unit at ask_price;
12             end
13         else
14             if the pending public order has traded then
15                 if private signal is demand then
16                     sell 1 widget in the private market;
17                 else if private signal is supply then
18                     buy 1 widget from the private market to cover the naked short
                         position;
19                 end
20                 unit ← units +1
21         end
22     end
```

Liquidity provider: a liquidity provider, often referred to as a market-maker or a dealer[11], is a common intelligent trader in financial markets. Formally, an intelligent trader is defined as a trader that utilizes system-wide information, i.e., the public order book, all public trades in the past, and trade volume. For simplicity, the two terms liquidity provider and market-maker are used interchangeably in this paper. The differences between the market-maker and ZITs are threefold: (1) the

[11]In practice, licensed designated market-makers have contractual obligations to post bid and ask orders to maintain a "fair and orderly markets", see https://www.nyse.com/market-model. We are interested in more generic non-obligated liquidity providers. For the purpose of this paper, the terms liquidity provider and market-maker are interchangeable.

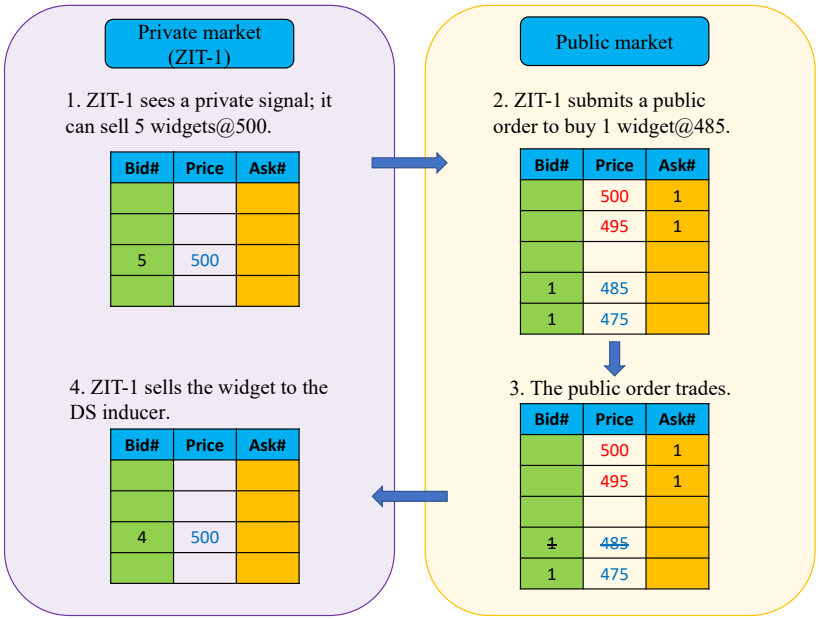


Figure 6.3. An example arbitrage trade by a Zero-Intelligent Trader. The DS inducer sends a private signal to buy from ZIT-1 five widgets at a price of 500 cents. ZIT-1 sends a public bid order of one widget at 485 cents. Imagine the order gets traded, meaning that ZIT-1 has paid 485 cents in exchange for one widget with another anonymous market participant. It then proceeds to the private market to sell the widget to the DS inducer. The arbitrage profit ZIT-1 receives is $500 - 485 = 15$ cents. ZIT-1 will continue this loop until all five units in the private order are consumed.

market-maker does not receive any private signals, meaning that it has no information regarding the underlying economy throughout the entire simulation, and (2) the market-maker does not know when the incentive structure (private signals) of ZITs will change, and (3) the market-maker is able to utilize system-wide information to discover the price of the widget whilst ZITs do not. Implementation of a market-maker varies, but the primary mission is to quote a bid-ask pair of the same underlying asset to capture the spread as its profit, condition on both orders to be traded. The submitted quote prices are typically calculated by some model that takes both fundamental and/or technical indicators as the state input.

By submitting a bid-ask pair, the market-maker is exposed to *inventory risk*. The market-maker does not know when the economic regime may shift and whether the regime will ever come back. Consequently, it is possible that only one side of the bid-ask pair is traded. In that case, the market-maker may accrue its inventory positions and end up with a net-positive (long) or a net-negative (short) inventory position.

Empirically, inventory risk is typically mitigated by two methods: (1) buy back (offload) the short (long) positions before a specific time (e.g., end of the trading day) from (to) other market participants[12] and (2) hedge the risk using alternative financial instruments (e.g., delta hedge). The former is common in less-liquid assets, while the latter is favored for liquid assets.

In this simulation, there is neither an inter-dealer market to frictionlessly offload inventory at the end of day (assumed in Glosten and Milgrom (1985) models and Kyle (1985) models) nor alternative financial instruments to hedge inventory risk. In order to mitigate inventory risk, the market-maker will need to account for inventory rebalancing dynamically while profiting from the bid-ask spread. Our market-maker constantly monitors the public order book. Every time the public order book changes, namely, an order submission or cancelation event, the market-maker computes the midpoint of the best bid and the best ask[13].

$$midpoint = \frac{best\_bid_k + best\_ask_k}{2} \tag{6.9}$$

Note that the midpoint does not necessarily change when the public order book changes. Upon collecting $K = 10$ midpoint values, the market-maker submits a

[12]Rebalancing strategies depend on factors like inventory size, market condition and whether market-makers have contractual obligations. However, market-makers typically conduct rebalance trades continuously while supplying liquidity to the markets (i.e., size of bid orders and ask orders are imbalanced).

[13]In the past several years, the midpoint of the National Best Bid and Offer (NBBO) has often been used as a reference price in liquidity orders in order to provide price improvement benefits to market participants, See, for instance, the NYSE midpoint Liquidity order https://www.nyse.com/publicdocs/nyse/markets/nyse-arca/NYSE_Arca_Order_Suite.xlsx and the NASDAQ midpoint Extended Life Order https://www.nasdaq.com/articles/nasdaq-takers-get-price-improvement-too-2020-08-06.

bid-ask order pair (one widget per order) to earn a spread, MM_spread, which is set to be a fixed scalar value of 10 (cents).

$$P_{MM}^{bid} = \frac{1}{K}\sum_{i=0}^{K} midpoint - \frac{MM\_spread}{2} \tag{6.10}$$

$$P_{MM}^{ask} = \frac{1}{K}\sum_{i=0}^{K} midpoint + \frac{MM\_spread}{2} \tag{6.11}$$

Once a bid-ask order pair is submitted, the market-maker clears its memory buffer and re-starts the midpoint bid-ask value collection process. In the meantime, the market-maker will also monitor its outstanding orders every five seconds. If its outstanding order sits too "deep" in the LOB on either the bid or the ask side, the market-maker will cancel the outstanding order and immediately submit an order to rebalance its inventory position[14]. More specifically, the market-maker monitors the following two conditions every five seconds:

$$Condition\ 1:\ P_{MM}^{bid} < best\_bid - \Delta P \tag{6.12}$$

$$Condition\ 2:\ P_{MM}^{ask} > best\_ask + \Delta P \tag{6.13}$$

We set the threshold $\Delta P = 100$ (cents). The rationale behind this inventory rebalancing strategy is that if the outstanding order sits too deep in the order book, it is possible that the market has permanently shifted from its current regime and the outstanding order is unlikely to be traded. By submitting a rebalancing order, the market-maker effectively decides to "reset" itself to adapt to the new public market regime.

We emphasize that the use of system-wide information is a crucial property that differentiates ZITs from a market-maker. In machine learning literature, a market-maker can be considered as a model-based learner where a model of the environment is built into the robot. A model-based agent uses environment observables as inputs, together with a built-in model of the underlying environment, to infer a hidden,

[14]Cross the spread to buy/sell at the best ask/bid price.

possibly dynamic state. Here, the state is the equilibrium price; learning is reflected by dynamically adapting to changing market regimes via filtering the LOBs in order to estimate the equilibrium price. See Algorithm 6.2 for the pseudocode of MM.

Demand-Supply inducer: the demand-supply inducer (DS inducer) is responsible for providing private signals to ZITs via designated private markets; it does not interact with the public market. At the beginning of each period, it sends one bid/ask order with five widgets to each ZIT privately. The price of each order is determined by the demand-supply incentive mechanism, which is outlined in the next section. The DS inducer has unlimited capital and widgets to fund all trades.

6.2.4 *Procedure*

We conducted four sets of simulations. In each set, we ran a session of twenty ZITs (called ZITs-only), and a session of twenty ZITs with one market-maker (called ZITs-with-MM). In each set, the signal sequence received by ZITs over periods in both simulations is exactly the same, and so is the theoretical equilibrium sequence. In the ZITs-only simulations, the economy comprises a DS inducer and twenty ZITs; in the ZITs-with-MM simulations, one market-maker is introduced to the economy. Each simulation consists of 100 periods, each period $m \in [0..100]$ lasts 60 seconds. On average, the last trade occurs 9.47 (SD = 1.48) seconds after the start of each period. All algorithmic traders start at the same time when both the public market and the private markets are open. At the beginning of period m, the DS inducer sends private signals to each ZIT. ZITs submit public orders immediately after receiving their private signals. Between period m and its next period $m + 1$, there is a cool-down stage in which all outstanding orders in both markets are canceled by the system. Notably, in ZITs-with-MM simulations, outstanding public orders (if any) submitted by the market-maker are **not** canceled; instead, its orders are carried over from period m to period $m + 1$. By forcing the market-maker to carry over public orders across periods, we enforce that the market-maker does not receive any information about the underlying economy and strictly follows its designed algorithm.

Algorithm 6.2: Market Maker

```
1  Function Main():
       /* initialisation of parameters                                    */
2      MM_spread := 10;
3      rebalance_limit := 100;
4      bound := 10;
5      midpoints := list();
6      do concurrently
7          ObservePublicMarket(MMS_spread, midpoints, bound);
8          CheckPendingOrders(rebalance_limit);
9      end
10
11 Function ObservePublicMarket(MM_spread, midpoints, bound):
12     if order book changed then
13         midpoints.add((best_bid+best_ask)/2);
14         if size(midpoints)=bound and no pending orders then
15             bid_price = 1/bound * sum_{i<=bound} midpoints[i] - MM_spread/2;
16             as_price = 1/bound * sum_{i<=bound} midpoints[i] + MM_spread/2;
17             submit buy order for 1 unit at bid_price ;
18             submit sell order for 1 unit at ask_price ;
19             midpoints = list() // clear the midpoints collection list
20         end
21     end
22
23 Function CheckPendingOrders(rebalance_limit):
24     sleep_time := 5;
25     best_bid := current best bid in the order book;
26     best_ask := current best ask in the order book;
27     if pending buy order and sell order traded then
28         bid_price = bid price of market maker's pending order;
29         if bid_price < best_bid − rebalance_limit then
30             cancel pending buy order;
31             submit buy order for 1 unit at best_bid;
32         end
33     end
34     if pending sell order and buy order traded then
35         ask_price = ask price of market maker's pending order;
36         if ask_price > best_ask + rebalance_limit then
37             cancel pending sell order;
38             submit sell order for 1 unit at best_ask;
39         end
40     end
41     sleep(sleep_time);
```

6.2.5 *Architecture*

We conducted all simulations using the Flex-E-Markets[15]. The Flex-E-Markets is a real-time centralized trading platform equipped to support both manual trading via a web-based graphical user interface (GUI) and algorithmic trading via an application programming interface (API). In all simulations, the GUI is merely for visualization purposes because no manual trading is involved; all tradings are fully automated using the APIs. See Figure A.2 in the appendix for a snapshot of the GUI. The platform is hosted on a server in the USA.

All algorithmic traders are scripted in Python and hosted on a second server in Australia. All communications are transmitted at the market level, meaning there is no local communication among traders within the server. For instance, when the DS inducer sends a private order to the trader robot[16], the communication does not happen within the Australian server; instead, a private order is first sent to the private market, and then the private market sends the signal back to the trader robot. On average, communication latency, or the time for a single message to reach the other end, is ~ 0.75 seconds due to the physical distance between the two servers (round-trip latency is ~ 1.5 seconds). See Figure 6.4 for an illustration of the simulation architecture and Figure A.5 in the appendix for an illustration of communication latency.

6.3 ANALYSIS FRAMEWORK

In this section, we report the detailed steps of our analysis. We first provide a brief overview of the dataset, then describe the methodology for each research question.

6.3.1 *Data*

The dataset for analysis consists of four parts: allocation, holding, order, and trade. The dataset has a depth similar to, if not deeper than, the NYSE Trade and Quote

[15]©Flex-E-Markets, Salt Lake City (UT), USA; see https://adhocmarkets.com/
[16]An order can be thought of as a message that contains a structured data.

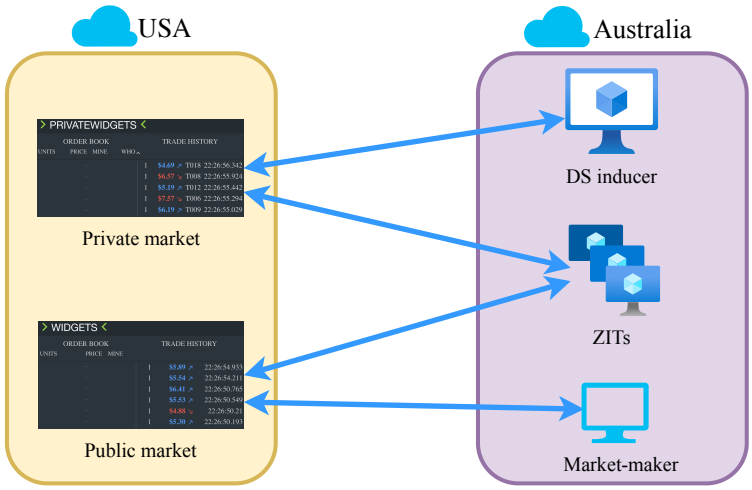


Figure 6.4. A real-time trading architecture. Both markets are hosted on a server that is physically located in the USA (yellow box). The algorithmic traders are hosted on a separate server that is physically located in Australia (purple box). Dual-way arrows in blue indicate the communication directions in the trading simulation.

(TAQ) database; in a nutshell, we know who submitted what order at what time. *Allocation* data are generated by a script that runs at the beginning of every simulation session. The data contain all private signals allocated to ZITs (see Subsection 6.2.1 for a detailed description of the economy and private signals). Holdings, orders, and trades data are generated by the Flex-E-Markets. *Holding* data contain cash and inventory position of all market participants in all periods. *Order* data provide access to the details of every single order, including its submission time, bid/ask side, price, units, and the identification of the order submission. *Trade* data consist of all historical trades; the level of details is similar to the order data. See Appendix Table A.11 for an example of the data. We will open-source all data and source code in a designated public repository.

6.3.2 *Economic efficiency and Pareto optimum*

We ask whether the economy will reach the Pareto optimal state (or Pareto efficiency) through trading activities and whether the introduction of a market maker improves the economic welfare. The Pareto optimal state is achieved when all widgets are distributed such that no one is better off without making any other agent worse off; as such, the total economic surplus is maximized, and the economy is said to yield Pareto efficiency (Arrow, 1951; Debreu, 1951, 1954; Mas-Colell, Whinston, & Green, 1995). See Figure 6.1 for an example demand and supply curve in our simulation.

For each period $m \in [0..100]$, we calculate the **actual surplus** generated through trading activities, AS_m, and compare that with the **theoretical surplus** under the Pareto optimal state, TS_m. By definition, it is obvious to see that $AS_m \leq TS_m$. If AS_m equals TS_m, then in period m the economy is said to reach the Pareto optimal state. Conversely, if AS_m is less than TS_m, then in period m the economy is said to reach a suboptimal state. To quantify how efficient the economy is in period m, we use the concept of **allocative efficiency**, the ratio $\frac{AS_m}{TS_m} \times 100\%$, (Smith, 1962; Gode & Sunder, 1993).

Each period m lasts 60 seconds (see Subsection 6.2.4 for details on the simulation procedure), which is sufficient for the economy to reach an equilibrium. The average time between the first trade and the last trade in each period m is around 10s (SD=2s). The research question is whether the ending equilibrium in period m is Pareto optimal.

We conjecture that the introduction of the market maker will improve Pareto efficiency. We calculate the median, mean, and the standard deviation of the ratio $\frac{AS_m}{TS_m} \times 100\%$ across 100 periods, and conduct a *two-sample t-test* between the ZITs-only sessions and the ZITs-with-MM sessions under the following hypothesis:

H_0 : two samples come from continuous distributions with equal means.

H_1 : two samples come from continuous distributions with unequal means.

In addition to the two-sample t-test which assumes normality, we perform a *Wilcoxon rank sum test* on the same data under the following hypothesis due to the concern of skewed data:

H_0 : two samples come from continuous distributions with equal medians.

H_1 : two samples come from continuous distributions with unequal medians.

Liquidity

Important to economic surplus and efficiency is the concept of **liquidity** in the public market as the public trades are the source of economic efficiency; the more public trades, the higher the economic efficiency. To examine the liquidity, we calculate three common liquidity measures of the trading activities in the public market: *trade volume*, *quoted bid-ask spread*, *effective bid-ask spread*.

Trade volume: Since the data contain the identification of each order and trade, we can classify trade counts into different categories. Specifically, we count the (1) total number of trades between ZITs and the manager robot in the private market, (2) the total number of trades among ZITs in the public market, and (3) total number of trades by the market maker in the public market.

Quoted bid-ask spread: Another common liquidity measure used in empirical studies is the quoted bid-ask spread. We take a snapshot of the order book per 0.01s; the snapshot window is chosen due to the high-frequency nature of the trading activities. We exclude the data in which only one-side quote is available (i.e., only bid no ask or vice versa) and all data in the idle stage after the last trade in each period to avoid artificially boosting the statistical power. For each order book snapshot, the quoted bid-ask spread is defined as:

$$\text{quoted bid-ask spread} = \frac{\text{best\_ask} - \text{best\_bid}}{(\text{best\_bid}+\text{best\_ask}) \ / \ 2} \tag{6.14}$$

Note the quoted bid-ask is strictly positive since the best ask is by definition greater than best bid; otherwise a trade will occur.

Effective bid-ask spread: Brunnermeier and Pedersen (2009) propose to use the absolute deviation between an asset's fundamental value and its transaction price

to reflect the actual bid-ask spread. Many empirical studies have used this concept as a proxy of market liquidity. However, the asset's fundamental value is typically unknown in the empirical dataset, and often proxies of the asset's fundamental value are required. As a result, in practice this measure is conceivably noisy. Here a proxy of the asset's fundamental value is not required as the equilibrium price is the asset's fundamental value. For each period $m \in [1..100]$, we define the effective bid-ask spread as follows:

$$\text{effective bid-ask spread}_{m,i} = \frac{|\text{equilibrium price}_m - \text{public trade price}_{m,i}|}{\text{equilibrium price}_m} \quad (6.15)$$

where transaction $\text{price}_{m,i}$, $i \in [1..N_{transactions}]$ is the price of each transaction in period m.

6.3.3 *Leptokurtosis and tail risks*

The second research question is whether leptokurtosis of price changes originates from the economic interaction (i.e., trading) in the market, even when a Gaussian distribution governs the changes in the fundamentals of the economy. We investigate this question by probing for public trade prices from two dimensions: (1) the tails of the price difference distribution only and (2) the entire price difference distribution. We apply the following procedure to both the ZITs-only sessions and the ZITs-with-MM sessions unless otherwise stated. For simplicity, we use the ZITs-only sessions as an example.

To begin with, we pooled all trade prices together, let

$$X = \{|p_{s,i+1} - p_{s,i}| \mid s \in [1..4], i \in [1..N_s]\}. \quad (6.16)$$

where $p_{s,i}$ is the price of trade i of seed number s and N_s is the number of public trades s of seed number s.

We adopt a procedure by Clauset et al. (2009) to examine whether the tail of the positive return follows a power-law distribution (i.e., if the distribution is heavy-tailed). Below we lay out the detailed procedures of the power-law test.

If the tail of the positive return probability distribution function (PDF) $pr(x)$ follows a power-law distribution as opposed to an exponential distribution, then the return PDF is heavy-tailed. Formally, we test the following hypothesis:

$$H_0 : \text{the tail of the PDF of } X \text{ follows a power-law distribution.}$$
$$H_1 : \text{the tail of the PDF of } X \text{ follows an alternative distribution.}$$

To test the hypothesis, we employed a three-steps procedure.

Step 1: model fitting. We define the *tail* of the positive return distribution as the collection of values greater than a lower bound x_{min}. Under the null hypothesis, we can parameterize the tail of the PDF $\forall x \geq x_{min}$ as the following:

$$pr(x|\alpha, x_{min}) = \frac{\alpha - 1}{x_{min}} \left(\frac{x}{x_{min}} \right)^{-\alpha} \tag{6.17}$$

where $\alpha_j > 1$ is the tail exponent, or the scaling parameter.

In order to obtain the best-fit parameters, $\hat{\alpha_j}$ and $\hat{x}_{min,j}$, we first specify a range of possible lower bound values between one and three standard deviations of the return PDF X. Then for each possible choice of x_{min} within the range, we estimate the best fit $\hat{\alpha}$ using the maximum likelihood estimation (MLE) method. A unique analytical solution exists:

$$\hat{\alpha} = 1 + n \left[\sum_{i=1}^{n} \ln \frac{x_i}{x_{min}} \right]^{-1} \tag{6.18}$$

where x_i, $i \in [1..N_{tail}]$ are the observed values such that $x_i \geq x_{min}$. See Clauset et al. (2009) for a detailed derivation of the maximum likelihood estimator.

Next, given the best estimated scaling parameter $\hat{\alpha}$, $\hat{x}_{min}$ is determined by choosing the value that gives the minimum Kolmogorov-Smirnov (KS) goodness-of-fit statistic:

$$\text{KS statistics} = \max_{x \geq x_{min}} |S_N(x) - P_N(x|\hat{\alpha}, x_{min})| \tag{6.19}$$

$$\hat{x}_{min} = \arg\max_{x_{min}} \Big(\text{KS statistics}\Big) \tag{6.20}$$

where $S_N(x)$ is the empirical cumulative distribution function (CDF) of the data in the tail, and $P_N(x)$ is the CDF for the power-law model that best fits the data in the tail. Statistically, KS statistics converges uniformly to zero according to the *Glivenko-Cantelli theorem* (Glivenko, 1933; Tucker, 1959).

Step 2: power law model goodness-of-fit test. The fitting process in step 1 does not yield how well the power-law model fits the tail; hence next step is to determine whether the null hypothesis is plausible from a goodness-of-fit test.

We create a substantial amount of ($N_{synthetic} = 1000$) power-law synthetic datasets using the best estimated scaling parameter $\hat{\alpha}$ and the best estimated lower bound $\hat{x}_{min}$ obtained in step 1. Each synthetic dataset has the same size as the empirical data. For each synthetic data, we re-fit data to the power-law model and obtain its KS statistics, $KS_{synthetic}$. The goodness-of-fit p-value, $p_{power-law}$, is calculated as the proportion of the synthetic KS statistics that are greater than the empirical KS statistics in step 1:

$$p_{power-law} = \frac{N_{KS_{synethic} > KS_{empirical}}}{N_{KS_{synethic}}} \tag{6.21}$$

We choose 0.1 as the critical value; if $p_{power-law}$ is smaller than the critical value, then the null hypothesis is rejected. In our analysis, the critical value is a conservative one, meaning that the null hypothesis is rejected if there is less than a 10% chance that the best empirical KS statistics are smaller than the synthetic KS statistics.

Step 3: compare with alternative distributions. Even if the null hypothesis is not rejected (i.e., $p_{power-law} > 0.1$), we should cautiously conclude that the positive return data follow a power-law distribution because there could be an alternative distribution that fits the tail of the distribution equally well or better. To evaluate

and rule out possible alternatives, we conducted a model selection procedure. We first fit the possible alternative distributions to the entire positive return dataset. Then we calculated the log-likelihood values in the tail of the distribution (i.e., $\forall x_i > x_{min}$) using the best-fitted parameters. The final steps are to use common model selection criteria (log-likelihood, Akaike and Bayesian Information Criteria) and perform a *non-nested Vuong likelihood ratio test* by Vuong (1989) on the following hypothesis:

H_0 : if test statistics $\in [-c, c]$, the two models are indistinguishable.

H_1 : if test statistics $> c$, the power-law model is preferred than the alternative model.

: if test statistics $< -c$, the alternative model is preferred than power-law model.

where $c > 0$ is the critical value from a standard normal distribution for some significance level. In our simulation, we choose $c = 1.96$ (at 5% significance level) as the likelihood ratio test critical value[17].

In addition to studying tail risks, we use the two-sample Kolmogorov-Smirnov (KS) test to examine if the entire positive return distributions with and without MM are significantly different. Formally, we test the following hypothesis:

H_0 : two datasets are from the same continuous distribution.

H_1 : two datasets are from the different continuous distribution.

The test statistics is the maximum absolute difference between the empirical CDF of $\{X_{\text{ZITs-only}}\}$ and the empirical CDF of $\{X_{\text{ZITs-with-MM}}\}$:

$$\text{KS statistics} = \max_x |\hat{F}(x_{\text{ZITs-only}}) - \hat{F}(x_{\text{ZITs-with-MM}})| \tag{6.22}$$

[17]In Clauset et al. (2009) appendix C, the authors also point out that one can further check whether the sign of the likelihood ratio statistics is trustworthy to be the indicator of model preference. This additional step involves calculating the probability that the measured log-likelihood ratio is not zero when in fact, the true value of the ratio is close to zero.

where $\hat{F}(x)$ indicates the empirical CDF. We choose 1.96 (at 5% significance level) as the critical value.

6.4 RESULTS

We report the results from two dimensions: (1) economic efficiency and Pareto optimality, and (2) leptokurtosis of the price difference. However, first we probe the validity of the market-maker in this experiment. The validity concerns the two postulations that we imposed on the market-maker: (1) the market-maker can profit from providing liquidity, and (2) the inventory balance is controlled reasonably well.

In Table 6.2 we report the market-maker's holdings difference between its ending account balance and beginning account balance. While the market-maker has earned positive cashflows in all four simulations, it concluded with a non-zero inventory difference in three simulations despite a built-in inventory control mechanism. We attribute this result to communication latency and the concurrent nature of traders' actions. Nonetheless, the excessive inventory does not affect our results below. We conclude that the market-maker has passed our validity check[18].

End − Start	**Seed 1**	**Seed 2**	**Seed 3**	**Seed 4**
Cash difference (¢)	7701	9383	6709	1121
% of total actual surplus	0.87	1.01	0.72	0.18
% of average ZITs profit	17.36	20.22	14.39	3.51
Inventory difference (widgets)	4	0	4	3

Table 6.2. Market maker's holdings difference. We subtract the market maker's ending account balance from its beginning account balance. Although an inventory-rebalance strategy was embedded in the design, the market maker still ended up with a positive inventory balance in some cases. We attribute the excessive inventories to the communication latency and concurrency. The positive inventory balance is insignificant and does not affect our conclusions.

[18]A positive difference in the inventory balance means that the market-maker has more inventory at the end than the beginning. In the worst case, the inventories are worth zero cents at the end, and the market-maker is still profitable. Moreover, while not vital to the main results, by juxtaposing Table 6.2 and Figure 6.6, we may conclude that the market-maker satisfies the definition of a high-frequency trader in Kirilenko, Kyle, Samadi, and Tuzun (2017): (1) large number of trades, (2) small end-of-day positions, and (3) frequently switch between net long/short position.

6.4.1 *Economic efficiency and Pareto optimum*

Figure 6.5 compares the average allocative efficiency between ZITs-only sessions and ZITs-with-MM sessions. Allocative efficiency is measured by the percentage of theoretical maximum economic welfare achieved via trading activities (see Section 6.3 for details). The cyan line in Figure 6.5 shows the maximum efficiency (100%), in which the economy is said to reach its Pareto optimum.

The result on economic efficiency is perplexing. Firstly, while the theoretical maximum efficiency can be almost attained in some periods (max: ZITs-only 99.10% vs. ZITs-with-MM 99.57%), on average, trading activities yield a tad less than the maximum efficiency level (100%).

Secondly, introducing a market-maker did not necessarily help gain more efficiency. In seed 1, the mean efficiency increased by about 12% when the market-maker was introduced to the market; in ZITs-only, the mean was 73.31% (SD: 13.97%) whereas in ZITs-with-MM, the mean was 85.71% (SD: 12.4%) (median: ZITs-only 75.8% vs. ZITs-with-MM 89.45%). The improvement in efficiency is statistically significant, which is confirmed by both the two-sample t-test ($p < 0.01$) and the Wilcoxon rank-sum test ($p < 0.01$).

In contrast, in seed 2 and 3, there were no significant improvements in economic efficiency. In seed 2, the mean difference in efficiency was merely 1.7%; in ZITs-only, the mean was 83.89% (SD: 11.50%) whereas in ZITs-with-MM, the mean was 82.24% (SD: 13.16%) (median: ZITs-only 85.51% vs. ZITs-with-MM 86.38%).

Akin to Seed 2, in Seed 3 the mean difference in efficiency is merely 2.8%; in ZITs-only, the mean was 85.69% (SD: 10.36%) whereas in ZITs-with-MM, the mean was 88.43% (SD: 10.23%) (median: ZITs-only 88.51% vs. ZITs-with-MM 91.49%). In both cases, the mean differences are imperceptible (Wilcoxon rank-sum test: $p_{seed\ 2} = 0.59$, $p_{seed\ 3} = 0.02$).

Perhaps even more egregiously, in seed 4, the market-maker deteriorated the economic efficiency by about 28%; in ZITs-only, the mean was 86.40% (SD: 8.51%) whereas in ZITs-with-MM, the mean was 58.40% (SD: 18.26%) (median: ZITs-only 87.72% vs. ZITs-with-MM 60.22%).

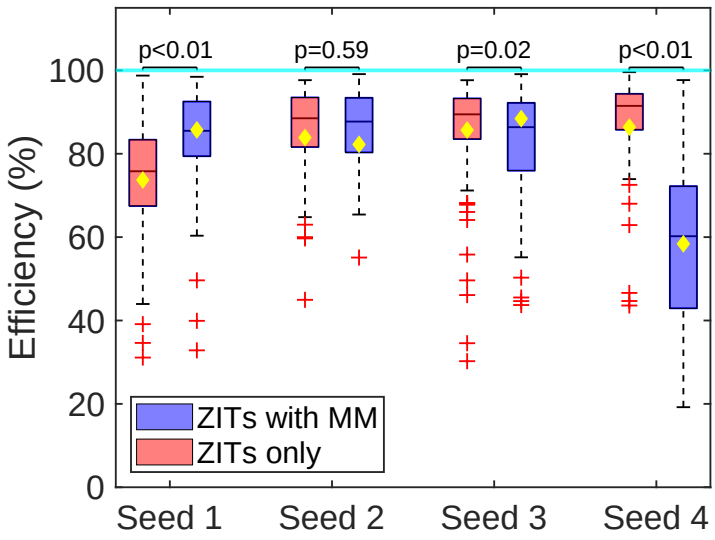


Figure 6.5. Allocative efficiency. We compare allocative efficiency when introducing the market maker to the economy versus when the market maker is absent from the economy. Allocative efficiency is measured by the percentage of theoretical maximum economic welfare attained via trading activities (see Section 6.3 for details). The cyan line shows the maximum efficiency (100%), in which case the economy is said to reach Pareto optimum. Each **seed** represents a set of two simulations where the sequences of the theoretical demand-supply curve is the identical. Each data point is the allocative efficiency of a single period $m \in [1..100]$. The results of ZITs only session are shown in red boxes, whereas the results of ZITs-with-MM are shown in blue boxes. Yellow diamonds show the equal-weighted average value of efficiency in each session. Black solid lines inside the bars show the median efficiency. P-values above the boxes are for the Wilcoxon rank sum test on whether the median efficiency is significantly different between ZITs-only sessions and ZITs-with-MM sessions. The efficiency levels are significantly different between ZITs-only and ZITs-with-MM in seeds 1 and 4, whereas we cannot differentiate the efficiency levels between the two in seeds 2 and 3.

Liquidity in the public market

To elucidate the mixed results in economic efficiency and market-maker's impact on economic efficiency, we further evaluate the *liquidity* in the public market because it is commonly believed that market-makers improve liquidity in the financial markets by tightening the bid-ask spread and inducing more trading activities. Inspired by

both theoretical and empirical market microstructure studies, we use three measures to proxy liquidity, (1) trade count, (2) quoted bid-ask spread, and (3) effective bid-ask spread. We refer readers to Section 6.3 for detailed definitions.

Trade count. Traders' identification in the dataset has enabled us to categorize all trades into three groups, (1) private trades between ZITs and the manager robot, (2) public trades between ZITs, and (3) public trades in which one party is the market-maker. Figure 6.6 illustrates the results.

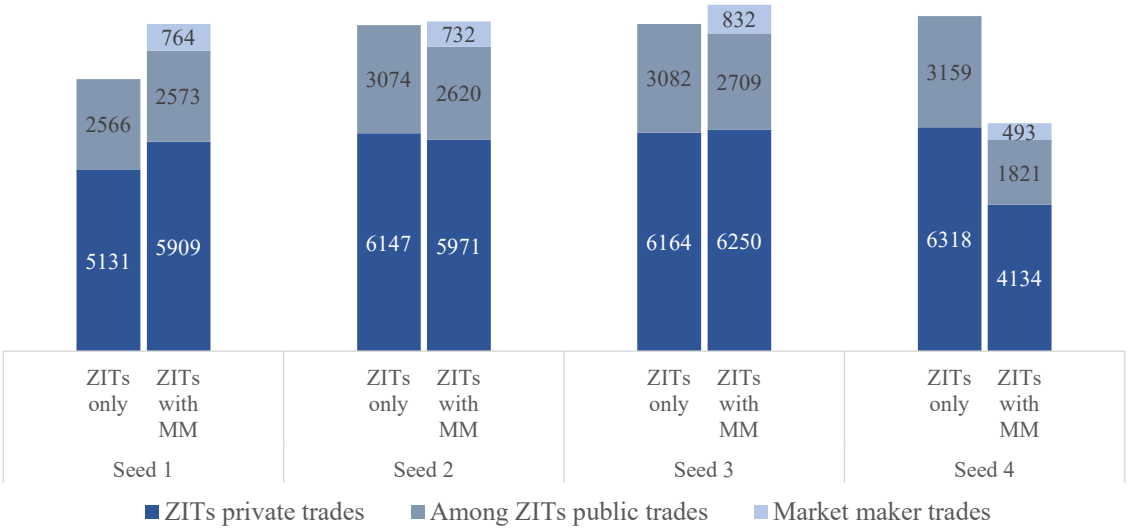


Figure 6.6. Trade count. In blue bars, we count the number of private trades in the private market. In gray bars, we count the number of public trades among the ZITs, i.e., the two parties involved in a transaction are both ZITs. In light blue bars, we count the number of trades by the market maker, i.e., one party involved in a transaction is the market maker and the other party is a ZIT.

The first observation from Figure 6.6 is that the pattern of the trade count is the same as the pattern of efficiency in Figure 6.5, which shows that more (fewer) trades lead to higher (lower) economic efficiency. The result indicates that more public trades will increase the number of private trades and thus generate a higher surplus.

The second observation concerns the market-maker's trading behavior. In seed 1, the number of public trades among ZITs remained the same, irrespective of whether the market-maker was introduced. As a result, the increase in total public trades

can be attributed to the market-maker's trades. In this scenario, the market-maker indeed supplied liquidity to the economy by facilitating more public trades through matching the demand and supply.

In seed 2 and 3, it appears that the market efficiency level persisted after the market-maker engaged, but the situation was slightly different from seed 1. Total public trades remained roughly the same while public trades among ZITs reduced. The market-maker acted like a high-frequency trader that took away some trades that could have occurred between two ZITs.

In seed 4, the total number of public trades plummeted. While the market-maker seemed to have inclinations to supply liquidity, the amount was a drop in the bucket. This observation elicits two questions: (1) why was there a dramatic reduction in efficiency, and (2) why was the market-maker unable to supply enough liquidity to address the reduction in economic efficiency?

To answer the two questions, we instantiate a particular period where the market-maker did not make any trade; we call such a scenario an *idle period*. Figure 6.7 entails two plots of public trade price formation dynamics. The two cases share an identical demand-supply curve and Pareto optimal equilibrium, yet the efficiency level dropped by 18.56% when the market-maker was introduced.

Figure 6.7 illustrates that the massive reduction in efficiency was due to a lack of public trades. The yellow shaded area of the ZITs-with-MM session, which depicts the bid-ask spread between the best bid and best ask, is much larger than the yellow shaded area of the ZITs-only session. Accompanied by a fewer number of trades (blue dots), the figure shows that the public market has experienced a liquidity drain in ZITs-with-MM sessions, despite having a liquidity provider in place.

Figure 6.7B shows that the market-maker was loath to supply liquidity. The market-maker submitted an ask order at the beginning of the period (the red dot). By design, the order was carried over from the previous period (and hence liquidity taking from the previous period); that is, the market-maker was prone to sell a widget since it had an excessive positive inventory at the start of this period. Constrained by inventory rebalance requirement, it had to wait until this order was traded before submitting new orders. Admittedly, new ask orders were coming in, e.g., an ask order of 215 cents at $T = 1.946$ seconds, which in hindsight should

trigger the rebalance function in theory. However, the ask sat at the order book for merely 0.3 seconds before it was consumed. The duration is far less than the five-second scheduled time to call the rebalance function; the duration is even less than the average message transition time between the two servers, meaning that by the time the market-maker received this information, the order had been consumed; the market-maker could not perform any action[19]. As a result, from the market-maker's perspective, its ask order was the best ask in the market throughout the entire period.

Quoted bid-ask spread. We define the quoted bid-ask spread as the ratio of bid-ask spread to mid-point of best bid and best ask, $\frac{\text{best\_ask} - \text{best\_bid}}{(\text{best\_bid}+\text{best\_ask}) \ / \ 2}$. For each period $m \in [0..100]$, we take the average, and collect 100 data-points for each session. We then pooled the data from all sessions together. See Section 6.3 for details.

Figure 6.8A portrays that the market-maker successfully tightened the bid-ask by 6.47% on average; in ZITs-only, the mean was 30.70% (SD: 9.97%) whereas in ZITs-with-MM, the mean was 24.23% (SD: 8.47%) (median: ZITs-only 30.08% vs. ZITs-with-MM 23.29%). The reduction in quoted bid-ask spread is statistically significant, confirmed by the two-sample t-test (p =7e-22) and the Wilcoxon rank-sum test (p =7e-23).

In Figure 6.8B, we regress the average quoted bid-ask spread of ZITs-with-MM sessions on the ZITs-only sessions. The slope coefficient is much smaller than 1 ($\beta_1 = 0.32$, p =1e-14), which indicates that the quoted bid-ask spread was lower in ZITs-with-MM sessions.

Using quoted bid-ask spread as a proxy of liquidity measure, we show that the market-maker has indeed improved the liquidity in the public market. This result is consistent with the consensus on the market-maker's impact on liquidity provision in market microstructure studies.

Effective bid-ask spread. Brunnermeier and Pedersen (2009) propose to use the absolute deviation between an asset's fundamental value and its transaction price to reflect the effective bid-ask spread in an attempt to find out *true trading cost*. Formally, the effective bid-ask spread is calculated as $\frac{|\text{equilibrium price}_m - \text{public trade price}_{m,i}|}{\text{equilibrium price}_m}$

[19]To be more precise, one could design a strategy to account for those high-frequency trades ex-ante. However, the strategy will become immensely complicated, which dilutes the purpose of the paper.

for each period $m \in [0..100]$ and each transaction i. We take the average for each period $m \in [0..100]$, and collect 100 data-points in each session. We then pooled the data from all sessions together. See Section 6.3 for details.

Figure 6.8C shows that the effective bid-ask spread remained roughly the same. The deviation is only 0.6% on average; in ZITs-only, the mean was 16.23% (SD: 5.64%) whereas in ZITs-with-MM, the mean was 15.14% (SD: 5.90%) (median: ZITs-only 15.67% vs. ZITs-with-MM 14.21%). Although the reduction is statistically significant (two-sample t-test p =9.3e-3; rank-sum test p =1.3e-3), the economic significance is small.

In Figure 6.8D, we regress the average effective bid-ask spread of ZITs-with-MM sessions on the ZITs-only sessions. The slope coefficient is also lower than one, but the effect of the market-maker on the effective bid-ask spread is not as strong as its effect on the quoted bid-ask spread.

Number of trades explains higher economic efficiency while the bid-ask spread does not.

The above results have already shed some light on the source of economic efficiency: a larger number of trades improves economic efficiency. However, while successfully tightening the bid-ask spread, the market-maker does not necessarily improve efficiency by inducing more trades. To firmly examine the relationship between economic efficiency, trade volume, and the bid-ask spread, we regress the difference in allocative efficiency against (1) the difference in quoted bid-ask spread, (2) the difference in effective bid-ask spread, (3) the difference in the number of private trades, and (4) number of market-maker trades. All differences are obtained by subtracting values of ZITs-only sessions from values of ZITs-with-MM sessions.

Figure 6.9 illustrates the regression results. The upper two plots (A&B) show that the correlation between the difference in efficiency and the difference in the bid-ask spread is inconclusive and noisy. The regression slope coefficient for difference in quoted bid-ask spread is 0.22 (p =0.03); the regression slope coefficient for difference in effective bid-ask spread is -0.78 (p =4e-4). In general, a negative slope coefficient is expected if a smaller bid-ask spread leads to a higher efficiency level.

Conversely, the bottom two plots (C&D) clearly illustrate that a larger economic efficiency requires more trades. In the bottom-left plot, the regression slope coefficient of the difference in private trades is 1.24 (p =1.28e-133), which indicates that more private trades are needed for all ZITs to be better-off. At the same time, to incentivize more private trades, more public trades are required. To see if the market-maker has made the economy better-off by inducing more trades, we shift our attention to the bottom-right plot. There, we observe that the regression slope coefficient is 1.46 (p =4e-9). Hence, it is obvious that economic efficiency is improved when the market-maker trades more.

Furthermore, from Figure 6.9D, we observe significant variability in the efficiency level when the market-maker conducted zero trades (i.e., the x-axis equals zero). This result echos Cliff (1997): the "apparent" demand-supply curve, which is based on the actual order submission price, is different from the theoretical demand-supply curve. Consequently, the apparent equilibrium is bound to deviate from the theoretical equilibrium unless traders' profit margins are strictly zero. See Appendix Figure A.4 for an illustration of a single period apparent demand-supply curve.

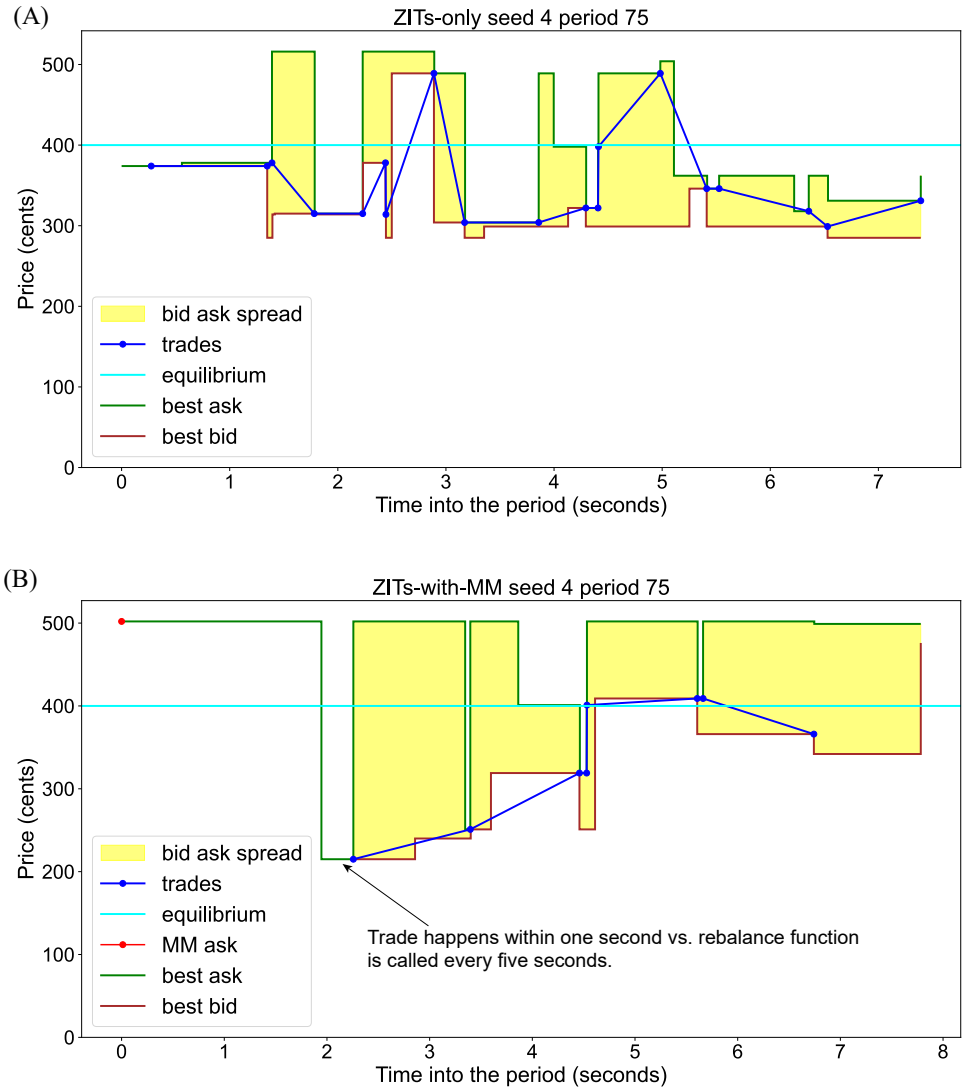


Figure 6.7. An example period where the market maker was idling. The theoretical demand-supply curves of the ZITs-only (**A**) and the ZITs-with-MM (**B**) plots are identical, yet the price formation dynamics are different, leading to different equilibrium and efficiency. The cyan line shows the *theoretical* equilibrium under the Pareto optimum. The green step curve depicts the best ask. The brown step curve depicts the best bid. The blue line and dots show the trade price time series. The yellow area within the best bid and the best ask curve illustrates the bid-ask spread whenever there is at least one order on both sides. **(B)**: The red dot at the beginning represents the market maker's ask order (sell one widget at 500 cents). This order was the first order in the period and the only order that the market maker had submitted throughout the period; we denote the time of this order as $T = 0$. At $T = 1.946$ seconds, another ask order (sell one widget at 215 cents) came in. As a result, the best ask decreased to 215 cents. At $T = 2.256$ seconds, a bid order (buy one widget at 215) was submitted, causing trade to occur at 215 cents. After this trade, the best ask returned to 500 cents.

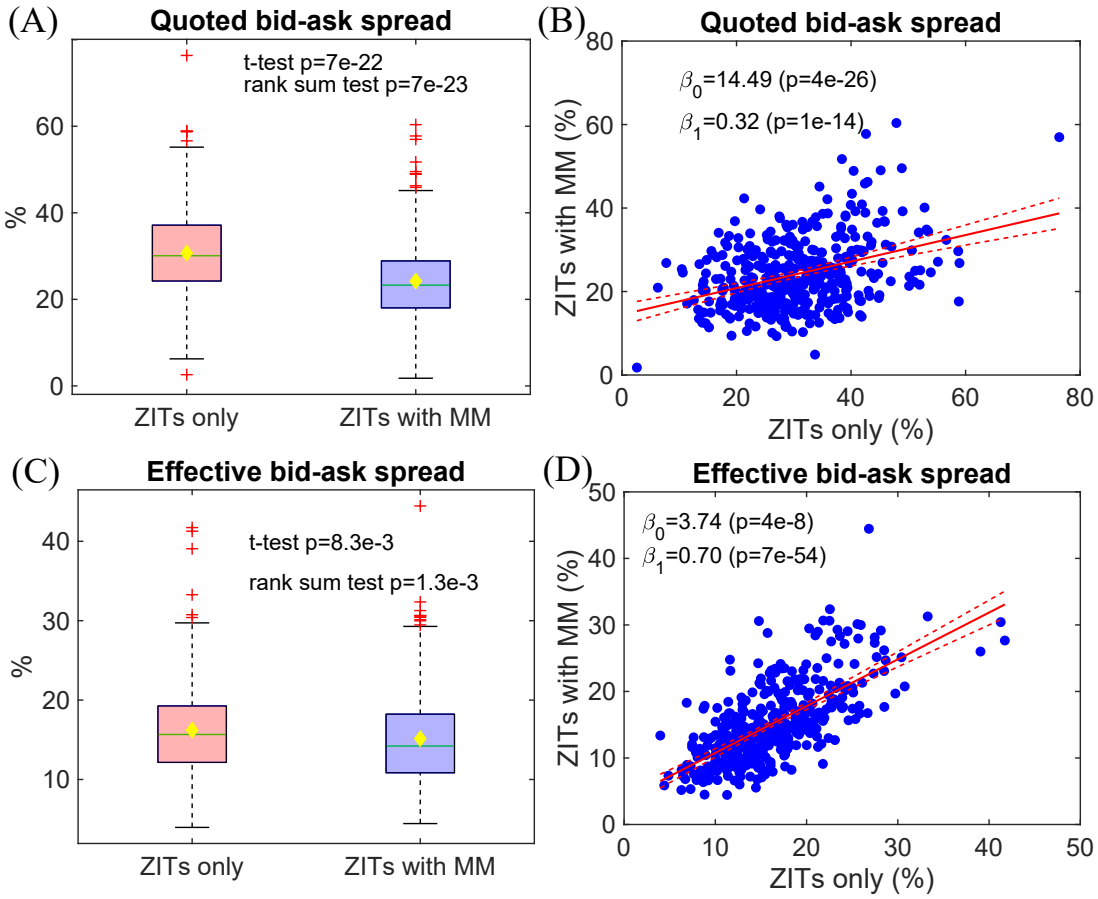


Figure 6.8. Average bid-ask spread. Quoted bid-ask spread is calculated as $\frac{\text{best\_ask} - \text{best\_bid}}{(\text{best\_bid}+\text{best\_ask}) \ / \ 2}$; the best bid and best ask quotes are obtained from orderbook snapshots. Effective bid-ask is calculated as $\frac{|\text{equilibrium price} - \text{public trade price}_i|}{\text{equilibrium price}}$. Section 6.3 describes methods in detail. **A & C.** The box plots compare the mean of the period-average bid-ask spread between ZITs-only and ZITs-with-MM sessions. Yellow diamonds illustrate the median. Green straight lines show the mean. We compare the mean values of two sessions using the two-sample t-test and the median values using the Wilcoxon rank-sum test. **B & D.** The relationship of bid-ask spread in ZITs-with-MM sessions and in ZITs-only sessions. Each blue dot represents the value of a single period m. Red straight lines correspond to fitted lines. Red dashed lines correspond to the upper and lower confidence bounds at the 95% level. Coefficients β_0 and β_1 are the intercept and the slope of the fitted line, respectively.

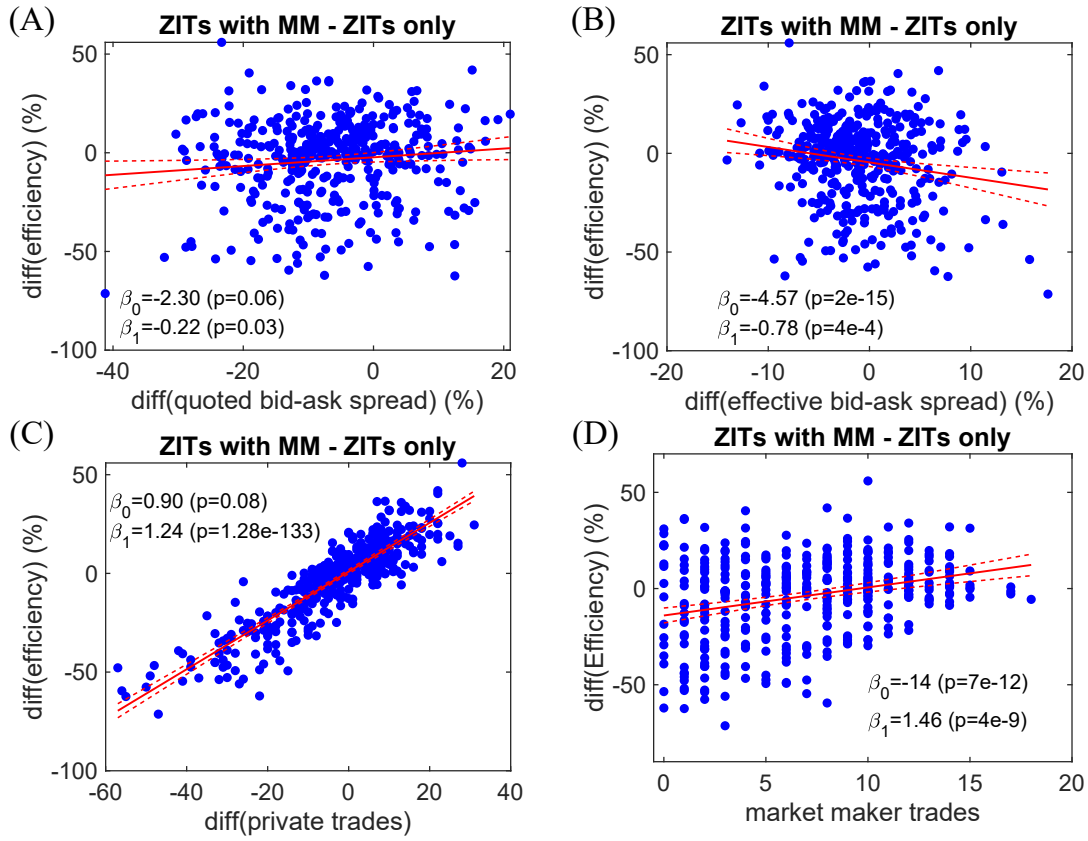


Figure 6.9. Difference in efficiency. In all four plots, the Y-axis is the difference in allocative efficiency between ZITs-with-MM and ZITs-only sessions (ZITs-with-MM minus ZITs-only). We plot the relationship between allocative efficiency and **(A)** the difference in quoted bid-ask spread, **(B)** the difference in effective bid-ask spread, **(C)** the difference in number of private trades and **(D)** the number of market maker trades. Each blue dot represents the value of a single period m. Red straight lines correspond to fitted lines. Red dashed lines correspond to the upper and lower confidence bounds at the 95% level. Section 6.3 describes methods in detail. Coefficients β_0 and β_1 are the intercept and the slope of the fitted line, respectively.

6.4.2 *Leptokurtosis and tail risks*

Table 6.3 reports the summary statistics of price difference data from all sessions. Although the underlying demand-supply curve and regime shifts are identical in both ZITs-only and ZITs-with-MM sessions, the kurtosis of the price difference is higher in ZITs-with-MM sessions (ZITs-only 4.08 vs. ZITs-with-MM 4.71).

$X = \{$**price diff**$\}$	**Mean**	**Std. dev.**	**Skewness**	**Kurtosis**	**Min**	**Max**
ZITs only	0.02	131.13	-0.02	4.08	-731	633
ZITs with MM	0.01	115.32	0.05	4.71	-643	813

Table 6.3. Summary statistics of the price difference dataset. Std. dev. stands for standard deviation. Kurtosis numbers are highlighted in red since they are the focus of our analysis.

To identify the source of a higher kurtosis, we utilize a framework by Clauset et al. (2009) to study if the tail of the distribution of absolute price difference, |price diff|, follows a power-law distribution $pr(x|\alpha, x_{min}) = \frac{\alpha-1}{x_{min}}(\frac{x}{x_{min}})^{-\alpha}$; if the power-law distribution fits the data better than the exponential distribution, the distribution is heavy-tailed (leptokurtic). See Section 6.3 for details on the power-law fitting procedure.

Figure 6.10 visualizes how well different models fit the tail of the distribution. We plot the empirical cumulative distribution function (CDF) of the absolute price difference (light blue dots) on a log scaled x-axis and y-axis. We choose the best fit parameter $\hat{x}_{min}$ as the starting point of the "tail of the distribution" (red vertical dashed lines). The figure shows that the power-law distribution (pink squares) fits the data better than the exponential distribution (blue triangles) and the generalized extreme value (GEV) distribution (black dots)[20].

The visual results are confirmed by the goodness-of-fit test and model selection criteria in Table 6.4. The p-values from the goodness-of-fit test are both one (GOF p =1), meaning that the power-law model is a good fit to the data in both cases[21].

[20]We used similar model comparison procedures to compare a group of known parametric distributions. The exponential distribution and the generalized extreme value distribution were the two that best fit the data.

[21]Note that the interpretation of the goodness-of-fit p-value is different from the usual statistical p-value. See Section 6.3 and Clauset et al. (2009) section 4.1 for details.

	ZITs only			ZITs with MM		
	Exponential	GEV	Power Law	Exponential	GEV	Power Law
GOF P	N/A	N/A	1.0	N/A	N/A	1.0
LogL	-2.14e3	-2.18e3	-1.28e3	-1.23e3	-1.22e3	-0.68e3
AIC	4.29e3	4.40e3	2.57e3	2.47e3	2.44e3	1.37e3
BIC	4.29e3	4.41e3	2.58e3	2.47e3	2.47e3	1.37e3
Vuong Z	15.62	15.63	N/A	11.48	11.46	N/A
Vuong P	3.85e-54	1.87e-54	N/A	9.40e-30	1.23e-29	N/A

Table 6.4. Goodness-of-fit and model selection criteria. GOF P corresponds to the p-value from the *goodness-of-fit test*, which quantifies the plausibility of a power-law fit. Interpretation of power-law p-value is different from the usual statistical p-value. If GOF P is close to 1, then the power-law model is a good fit to the data. See Section 6.3 and Clauset et al. (2009) section 4.1 for details. GEV stands for generalized extreme value distribution. LogL stands for log-likelihood value. AIC and BIC correspond to Akaike and Bayesian information criteria, respectively. Vuong Z corresponds to the test statistics from the non-nested Vuong likelihood ratio test. Vuong P corresponds to the p-value from the non-nested Vuong likelihood ratio test. N/A stands for not available. Each number highlighted in red demonstrates that its corresponding model is better under a particular model selection criteria.

Moreover, the power-law model is preferred over the exponential and the GEV model based on all three criteria, log-likelihood, Akaike, and Bayesian Information. The statistics of the non-nested Vuong likelihood ratio test are much larger than zero (ZITs-only 15.62, ZITs-with-MM 15.63) with p-values way below zero (ZITs-only p =3.85e-54, ZITs-with-MM p =1.87e-54), indicating that power-law model is preferred to the other two models. Hence, we conclude that the power-law distribution fits the tail of the distribution much better, and the distributions in both sessions are heavy-tailed.

Compare leptokurtosis in ZITs-only sessions with ZITs-with-MM sessions

Although the above procedure has enabled us to determine an appropriate distribution to fit the price difference data, the increase in kurtosis is yet to be scrutinized. Now that the power-law model is an appropriate model, we compare and interpret the best fit parameters, $\hat{\alpha}_{\text{ZITs-only}}$ and $\hat{\alpha}_{\text{ZITs-with-MM}}$.

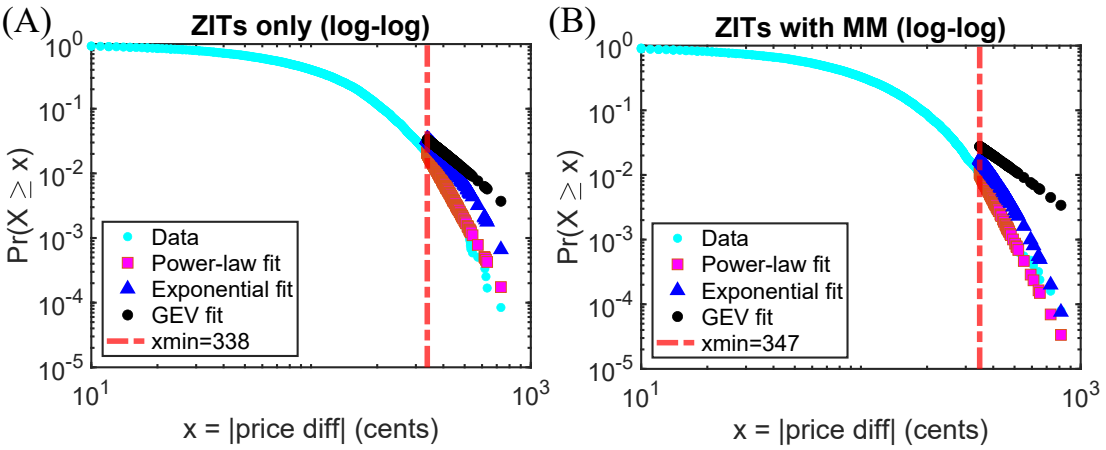


Figure 6.10. Distribution fitting of |price difference| on a log-log scale. In both figures, we plot the empirical cumulative distribution function (CDF) of the absolute public trade price difference. We calculate the best fitted power-law parameters of $\hat{\alpha}$ and $\hat{x}_{min}$ such that the power-law probability distribution function (PDF) $pr(x|\alpha, x_{min}) = \frac{\alpha-1}{x_{min}}(\frac{x}{x_{min}})^{-\alpha}$ best fits the empirical data. We then plot the best fitted values in pink squares. Similarly, we plot the best fit exponential distribution in blue triangles and the best fit generalized extreme value distribution (GEV) in black dots. The red vertical dashed line depicts the best fitted parameter $\hat{x}_{min}$, where we deem as the beginning of "the tail of a distribution". See Section 6.3 for detailed methods. Note that both the X-axis and the Y-axis are plotted on a log-log scale.

To obtain the variability of the power-law alpha, we bootstrap each dataset with replacements 1000 times, and estimate the best fitted alpha for each bootstrap. See Section 6.3 for details. Figure 6.11A shows that $\hat{\alpha}_{\text{ZITs-with-MM}}$ is greater than $\hat{\alpha}_{\text{ZITs-only}}$ by 0.6 (mean: $\hat{\alpha}_{\text{ZITs-with-MM}} = 7.25$, SD $= 0.35$ vs. $\hat{\alpha}_{\text{ZITs-with-MM}} = 7.85$, SD $= 0.63$; median: $\hat{\alpha}_{\text{ZITs-only}} = 7.24$ vs. $\hat{\alpha}_{\text{ZITs-with-MM}} = 7.81$). The difference is statistically significant, confirmed by both the two-sample t-test and the Wilcoxon rank-sum test. In Figure 6.11B, we visualize the interpretation of a higher alpha in the context of power-law distribution. Importantly, the tail of the probability density function does not necessarily become heavier while the density at the beginning becomes much larger. Hence, we attribute the increase of kurotsis to a higher power-law alpha.

To further substantiate our conjecture, we plot the empirical probability density function (histogram) of the absolute price difference data, |price diff|, in Figure 6.12. In Figure 6.12B, we zoom in on the tail of the histogram (for values greater than ZITs-only $\hat{x}_{min} = 338$). As the figure illustrates, there is hardly any difference, if not less, in the density mass of the tail in the ZITs-with-MM sessions. The decisive evidence to support our claim sits in Figure 6.12A. When we look at the entire histogram, it is crystal clear that the density mass in the middle of the histogram is smaller, whereas the density at the beginning becomes much more prominent in the ZITs-with-MM sessions (cyan histogram) as opposed to the ZITs-only session (grey histogram). The Kolmogorov-Smirnov test confirms that the difference between the two histograms is stark.

Juxtaposing Figure 6.11 and Figure 6.12 together, we conclude that the higher kurtosis of the price difference data is indeed due to trading activities from the market-maker. Intuitively, the market-maker "stabilized" the market by diminishing the movement of the prices between two trades, thereby reinforcing leptokurtosis in the price difference distribution.

6.5 DISCUSSION

We proposed a paradigm comprising a single-widget economy and a continuous open-book market to study economic welfare and tail risks. We demonstrated how

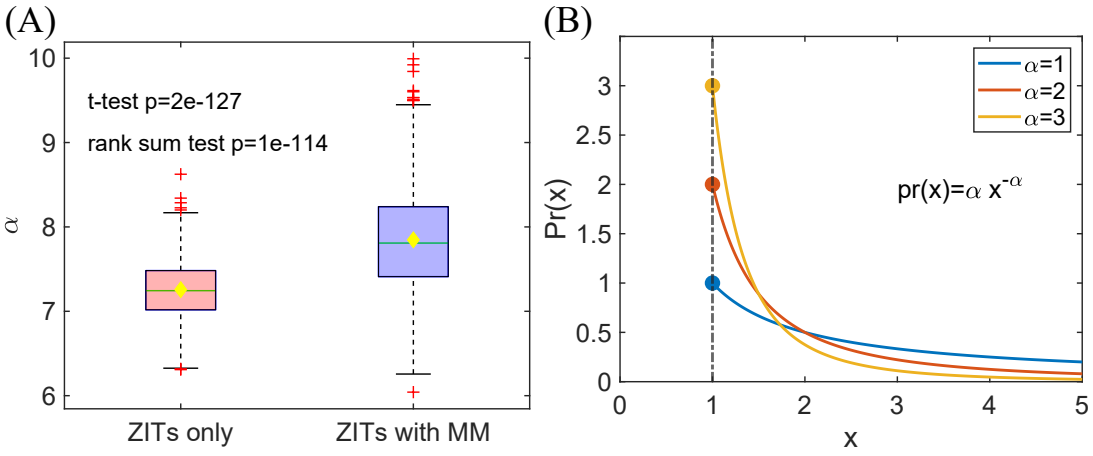


Figure 6.11. Bootstrap power-law alpha. We bootstrap the absolute price difference dataset with replacements 1000 times. For each bootstrap dataset, we calculate the best fit power-law parameters $\hat{\alpha}$ and $\hat{x}_{min}$. **(A)** The box plot compares the best fitted $\hat{\alpha}$ values of ZITs-only sessions against the best fitted $\hat{\alpha}$ values of ZITs-with-MM sessions. Yellow diamonds illustrate the median. Green straight lines show the mean. We compare the mean values of two sessions using the two-sample t-test and the median values using the Wilcoxon rank-sum test. See Section 6.3 for detailed methods. **(B)** We plot the power-law probability density function (PDF) with three different alpha values to illustrate the interpretation of a larger power-law α.

trading activities among market participants with various levels of intelligence affect economic welfare and leptokurtosis.

The paradigm stems from experimental economics and agent-based modeling. At the outset, the basic model for an agent in this paradigm is zero-intelligence, represented by traders who randomly place bid or ask orders (Gode & Sunder, 1993; Cliff, 1997). We contribute to zero-intelligence literature by formally defining the term *intelligence* as the ability to trade on system-wide information, such as trade price and volume. This definition connects the notion of zero-intelligence with literature on *swarm intelligence*, where intelligence emerges at a group level even when individuals have no access to system-wide information (Garnier et al., 2007; X.-S. Yang et al., 2016). We demonstrated an example of an intelligent agent, a market-maker.

Researchers in economics and finance have advocated coupling market microstructure and asset pricing studies as they are inseparably intertwined (O'Hara, 2003;

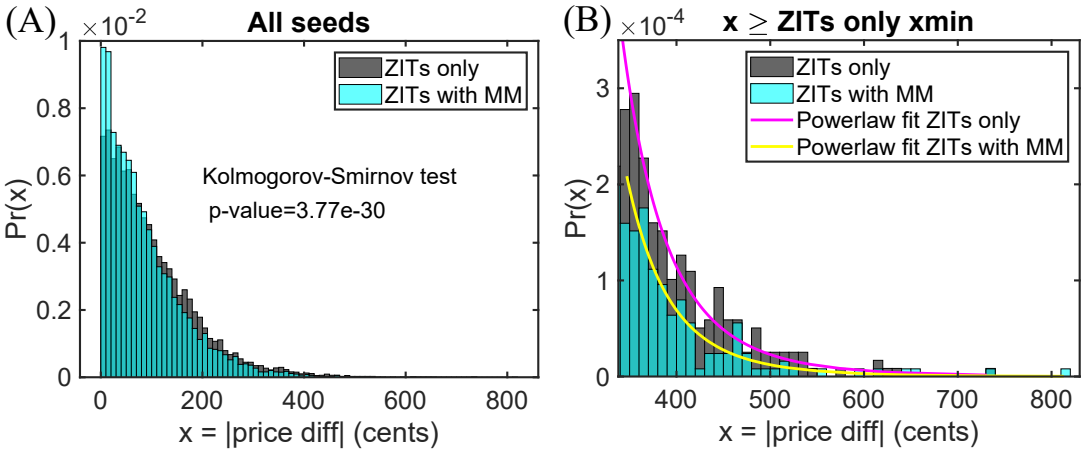


Figure 6.12. Histogram of |price difference|. **(A)** The plot illustrates the histogram of the absolute price difference for both ZITs-only (in black) and ZITs-with-MM (in cyan) sessions. We perform a Kolmogorov-Smirnov test on whether the two empirical probability density functions are statistically significantly different. **(B)** The plot depicts the tail of the histogram (i.e., values that are greater than ZITs-only $\hat{x}_{min} = 338$, which is calculated by fitting the power-law distribution to absolute price difference data of the ZITs-only sessions).

Easley & O'Hara, 2003). In this research, we unveiled that it is possible to connect the two by utilizing this paradigm. For instance, one puzzle between theoretical and empirical market microstructure is the *fundamental value* of a security. Theorists rely on fundamental values to develop models, whereas empirical studies need proxies to measure fundamental values when studying empirical data. However, some proxy measures are "admittedly crude" and often lead to measurement errors (Easley & O'Hara, 2003). Conversely, in our case, the fundamental value of a security is precisely the competitive equilibrium price that maximizes economic welfare, based on the first welfare theorem.

In contrast to prior studies, we found that a non-obligated market-maker is not innocuous to market welfare (Jovanovic & Menkveld, 2016; Menkveld & Zoican, 2017; Brogaard & Garriott, 2019; Aldrich & López Vargas, 2020). Two assumptions are central to a vast majority of market microstructure models (such as the Glosten and Milgrom (1985) model, the Kyle (1985) model, and their variants): (1) the true price of a security is revealed at some terminal point, and (2) market-makers can

frictionlessly offload all accumulated inventories at the true price without incurring any additional cost (Bouchaud, Bonart, Donier, & Gould, 2018). We showed that when relaxing these assumptions, a non-obligated market-maker has to become a liquidity taker sporadically due to its dynamic inventory risk management requirements. As a result, it could also be detrimental to the market welfare by draining the liquidity from the market.

Consistent with our hypothesis, introducing a profit-seeking liquidity provider increased leptokurtosis of the return distribution. Surprisingly, however, the increased kurtosis was not due to more tail risks but to the market-maker's intention to stabilize the market. By quoting a tight bid-ask spread, the market-maker effectively compressed the price changes within a finite bandwidth. Consequently, the density mass around one to two standard deviations of the price change distribution was taken away and clustered in the center, thereby enhancing leptokurtosis. The results on leptokurtosis have partially answered the question as to where the unique but ubiquitous feature in the price change distribution comes from (Farmer & Lillo, 2004). Financial econometrics literature typically equates leptokurtosis with tail risks (Orlowski, 2012; Kelly & Jiang, 2014), yet we showed that leptokurtosis could emerge in absence of any tail risk event. Hence, the results indicate that it is crucial to distinguish tail risks from leptokurtosis.

More broadly, this paradigm offers a unified *environment-agent* framework for scholars to research the contemporary financial system. The system contains numerous market mechanisms and market participants, such as humans, algorithmic traders, and machine learning algorithms. As the system becomes immensely complicated, it is difficult for researchers to identify the root source of an observed phenomenon. One advantage of this paradigm is its flexibility to include specific market structures and agents with various levels of sophistication by design, enabling researchers to disentangle one pattern from the others. Such flexibility should inspire researchers, particularly those who use experimental methods, to study topics like human-robot interaction in markets (March, 2019; Asparouhova et al., 2020; Bao et al., 2021).

The paradigm also connects researchers who study applications of artificial intelligence (AI) in finance with those who analyze markets at a granular level. With

the prevalence of AI spanning multiple disciplines, both academics and practitioners have expressed a significant amount of interest in deploying machine learning algorithms and AI agents in financial markets[22]. Despite the hype, whether and how AI traders may impose any influential impact on the financial system remains a genuine concern[23]. This paradigm blends well into the AI ecology where the concept of environment and agent constructs the central pillar, allowing the two fields to communicate with each other under the same framework.

To illustrate the potential of this paradigm in research, we sketch out two example implementations for future research. (1) We can extend the market-making algorithm to state-of-the-art machine learning algorithms, thereby allowing us to study the behavior pattern of AI agents and, in turn, examine how markets are affected by AI agents on an order-by-order basis. (2) We may also include extrapolative traders and market-makers to see if agents indeed are responsible for initiating, fueling, or abating asset bubbles/flash crashes (Barberis, Greenwood, Jin, & Shleifer, 2018).

Research is not the lone rationale for utilizing this paradigm; this paradigm complements the current policymaking process in the financial system. While regulators like the U.S. Security Exchange Commission (SEC) have adopted pilot studies (known as the "pilot program"[24]) to acquire information on potential implications of new policies, whether benefits outweigh costs remains a polemic (Harris, Kahn, McDonald, & Spatt, 2021). One contention argues that the benefits of running pilot programs are not obvious as many hypotheses are ambiguous, and the pilots seem "exploratory"[25]. On the flip side, policy trials on actively running markets may implicitly lead to unquantifiable social costs and irreversible consequences on market

[22]Financial institutions like JP Morgan have published a series of papers on machine learning in finance from a quantitative finance perspective, e.g., Ganesh et al. (2019) and Amrouni et al. (2022). Whether and how institutions have adopted machine learning algorithms remain unknown to the public. On the other hand, researchers in empirical asset pricing use machine learning algorithms to improve existing models, e.g., Gu et al. (2020) and Leippold et al. (2022).

[23]e.g., see the conversation article on human versus AI in financial markets https://theconversation.com/humans-v-ai-heres-whos-better-at-making-money-in-financial-markets-174937.

[24]The U.S. SEC trials new policies on a selected group of market participants/securities.

[25]United States Court of Appeals for the District of Columbia Circuit, New York Stock Exchange LLC et al., v. Securities and Exchange Commission, June 16, 2020, https://www.govinfo.gov/app/details/USCOURTS-caDC-21-01101/summary.

participants since companies and institutions cannot opt in or out of the pilots[26]. The paradigm sets an ideal platform for regulators to trial new policies in a flexible but controlled environment; it could help policy-makers to gain extra information without incurring high social costs. Notably, other fields have employed similar experimental paradigms as a compulsory procedure prior to policy implementation, particularly policies of responses to catastrophic events. For instance, *ex-ante* simulations are of paramount importance in construction resilience against earthquakes since assessing policy effectiveness *post* earthquakes results in unbearable monetary and social costs (Cimellaro, Noori, Kammouh, Terzic, & Mahin, 2022).

[26]For instance, the "tick size pilot" was conducted for merely three years before being terminated by the SEC, https://www.sec.gov/news/public-statement/tm-dera-expiration-tick-size-pilot.

Part VI

EPILOGUE

7

CONCLUSION AND FUTURE DIRECTION

In this dissertation, I studied learning efficiency of artificial and human agents in the environment of financial markets where tail risk are frequent and salient. To start, I crystallized some definitions and concepts in artificial intelligence and machine learning, and formally defined the term *intelligence* & *learning*. I then formally introduced reinforcement learning, one of the most successful frameworks that model artificial intelligence.

After establishing the theoretical foundations, I showed that both classical RL and distributional RL algorithms fail when rewards (returns) are affected by tail risk, i.e., leptokurtosis, in a stochastic bandit task. Inspired by classical statistics, I extended the distributional RL and the expected reward RL model to account for tail risk and efficiently estimate the state-action values. The resulting algorithm learns much faster and is robust once it settles on a policy.

Possible efficient gain for AI agents prompted us to wonder if humans exhibit the same kind of efficiency upon tail risk. To investigate this conjecture, I designed a behavioral experiment of stochastic bandit task where the stochastic rewards are drawn from Gaussian, Student-t and Exponential distribution. While I found substantial heterogeneity among participants, overall human reward learning appears to be guided by a desire to be efficient. This also translates into enhanced choice confidence, except when participants exhibit confusion as to the true form of the most efficient estimator.

Finally, I pondered over whether leptokurtosis is solely the consequence of the interaction of agents through the continuous open-book system. I approached this question by proposing a paradigm comprising a single-widget economy, a continuous

open-book market, and a group of zero-intelligence agents. I demonstrated that in the absence of a liquidity provider, trading generated leptokurtosis and hence excessive tail risk. Introducing a profit-seeking liquidity provider further increased leptokurtosis, but the tail risk was not worsened.

Overall, I wish to convey three key takeaways from this thesis. Firstly, we should recognize that agents' learning and decision-making are distinctly different upon non-Gaussian rewards than in a Gaussian environment; the latter is often assumed in most animal behavior studies. Here I focus on *tail risk*, invigorated by a reward shape that exists not only in financial decision-making in financial markets but also in natural calamities. Hence, when postulating about the environment, one should consider more generalized ambient factors, such as non-Gaussian reward functions. Secondly, while we show that there are efficiency concerns among humans, verging on statistical efficiency, more research is required to unveil the rationale behind those efficient learning. Thirdly, a new paradigm is required to understand how stylized effects emerge in financial markets (e.g., leptokurtosis). With more AI and ML market participants in the financial market, it becomes vital to understand agents' behavior at both the individual and the market levels.

7.1 THE ROAD AHEAD

The dissertation presented here exemplifies the potential of coalescing computer science, animal behavioral studies, and economics & finance for a common goal of building artificial general intelligence (AGI). As with other monographs, this research is a prelude to more research questions. Below, I discuss three exciting future directions.

Tail Risk, Prospect Theory and Multi-Armed Bandits

In this dissertation, I did not consider asymmetric treatment of potentially rare gains and losses, namely the *prospect theory* (Kai-Ineman & Tversky, 1979). It is widely documented that such could have a significant impact on human perception of tail risk and their decision-making under the conventional utility paradigm, but

not the reinforcement learning paradigm until recently, e.g., Shen, Tobia, Sommer, and Obermayer (2014); Prashanth, Jie, Fu, Marcus, and Szepesvári (2016). One potential future research topic is to conduct experiments that introduce asymmetry in the heavy-tailed rewards and study the speed of learning by both efficient algorithms and humans from both the negative and the positive end.

Moreover, while we elicited from the restless bandit task that human participants generally showed a tendency towards efficient estimation of gains, the general applicability of this result is limited by the restrictions of the two-arm bandit setting. In particular, in most real-life applications there is a high number (if not infinite) of underlying states. Under the utility paradigm, prominent studies have found that some theories on human agents' preferences hold only in two-outcome lotteries (Tversky & Kahneman, 1986; M. Cohen & Jaffray, 1988; Andreoni & Sprenger, 2011); agents often fail to understand whether they are in the state that they have already visited, or show preferences to lower complexity lotteries. Hence, a new set of theories and models have been proposed recently, attempting to address this myth for lotteries with more than two outcomes (Fudenberg & Puri, 2022; Puri, 2022). It would be important to design follow-up experiments to see if statistical efficiency preserves upon k-armed restless bandits ($k > 2$) and shed light on what type of information filtration forms the perceived notion of the state by agents.

Concern of statistical efficiency in model-based learning and meta-learning

Possessing the ability to deal with stochastic rewards (commonly referred to as reward uncertainty) is critical to building an AGI. We extend the expected reward research (e.g., Mahadevan (1996) and Dewanto, Dunn, Eshragh, Gallagher, and Roosta (2020)) by utilizing concepts of statistical efficiency. Our approach falls into the model-based learning category where agents construct hidden variables of the environment for learning and decision-making using models (Daw et al., 2011). The majority of the recent model-based reinforcement learning focused on state transition modeling $(s_t, a_t) \to s_{t+1}$, reward modeling is seemingly left behind even though the reward function r_t is an equally important member of the *experience* (s_t, a_t, r_t, s_{t+1})[1]

[1]States: $s_t, s_{t+1} \in S$; actions: $a_t \in A$; reward function r_t.

(Collins & Shenhav, 2021), if not the only component that matters (Silver et al., 2021).

While rewards are clearly critical in the reinforcement learning framework, the "reward-is-enough" hypothesis is controversial (Sajid et al., 2021; Vamplew et al., 2022). Forming expectations constitute the foundation of planning and prediction on all facets of our daily tasks, not just reward; humans seem to be very good at rapidly distilling information from ambient factors in the environment and forming some expectations of the future without explicit reward signals. This phenomenon has prompted scholars to unveil the mechanisms behind such efficiency in planning.

Recent advances in computational neuroscience provide a plausible, model-based conjecture by postulating that humans can learn in the absence of explicit reward through a model of surprise or uncertainty minimization (Sajid et al., 2021). This is the *predictive coding and active inference*[2] framework: agents perform perception inference or state-estimation through generative models prior to action selection (Friston et al., 2016; Parr, Pezzulo, & Friston, 2022).

From the perspective of efficiency, the goal of active inference is to make Bayesian probabilistic belief updating computationally manageable, by iteratively minimizing the expected Kullback-Leibler divergence between an approximated posterior distribution and the true posterior distribution with respect to agents' actions. Standard Bayesian learning is known to be computationally heavy, and the active inference framework simplifies this computationally hard problem via an efficient and dynamic approach.

Nonetheless, irrespective of model-based reinforcement learning or active inference, the bottom line is to speed up learning processes. Whilst the active inference framework aims at improving computational complexity, model-based statistical methods attempt to improve sampling efficiency.

Indeed, these frameworks have led to a prominent question: how do humans *meta-learn*? That is, how do humans learn to quickly adapt to new circumstances[3]? This question is critical to understanding human efficiency in decision-making and

[2]Active inference is a generalization of predictive coding in the control space (H. Brown et al., 2011).

[3]There is also another dimension whether they are aware of (expecting) regime shift or not, see Bossaerts (2022).

building an AGI (Naik & Mammone, 1992; Schmidhuber, 2007; Bengio, Bengio, Cloutier, & Gescei, 2013; Finn, 2018). In the second study, we observed that some subjects learned quickly to become efficient when they were aware that the environment regime had shifted, yet the mechanism behind their efficient adaptation is unclear.

Recent studies have mainly suggested two routes to approach this question. One conjecture is *transfer learning*, where a trained (parametric) model or representation of the sensory inputs in one task is transferable to other tasks, enabling agents to learn[4] faster in the new tasks with fewer training samples (Caruna, 1993; Torrey & Shavlik, 2010; Weiss, Khoshgoftaar, & Wang, 2016; Finn, 2018). In the context of the active inference framework, meta-learning emerges if agents are endowed with correct priors learned from other tasks (Sajid et al., 2021). This approach has been successful in representation-heavy tasks, e.g., mega vision and language models (Dai & Le, 2015; Radford, Narasimhan, Salimans, Sutskever, et al., 2018; Devlin, Chang, Lee, & Toutanova, 2019), and more recently in control problems (Reed et al., 2022). Pre-training these models requires distilling critical components from an enormous amount of information, a process where I conjecture that statistical efficiency could play an important role.

The other research scheme suggests an alternative, model-based mechanism for the learning rate itself. The rationale behind this route is that fast adaptation is also observed in tasks where environmental impact is minimized. The exact mechanism that steers the variation of learning speed remains obscure. One possible conjecture suggests that the prediction error signal in the anterior insular modulates the learning rate (model reference adaptive control) (d'Acremont et al., 2013; Bossaerts, 2018). Learning speed is adjusted with the intent to minimize surprises via a model, and hence it is a model-based meta-learning approach. Under the statistical efficiency framework, there could also be a separate mechanism that guides agents to switch between different models.

[4]Here *learning* is reflected by fine-tuning network weights.

Multi-agent systems in a market setting: interaction, collaboration and competition between humans and artificial agents.

The financial market, or even the economy in general, can be viewed as a multi-agent environment, populated with humans, robots and institutions (Shoham & Leyton-Brown, 2008). All agents have different views and incentives; their interaction, competition and collaboration contribute to everything we observe in the markets. Now in the era of AI we may soon witness new participants, artificial intelligent traders[5].

Despite a long history of human and robots participants trading simultaneously in a market setting, research of a multi-agent economy where both humans and artificial agents are involved is still at an inchoate phase. Traditionally, much research of multi-agent system, stemming from interests in how artificial agents interact, was studied through the lens of game-theory and computational algorithms (Shoham & Leyton-Brown, 2008). However, the notion of "agents" in economics & finance was limited to artificial agents whereas complexity of human agents are largely neglected. This trend continues: behavioral economics/finance and research in automated trading bots are still almost exclusively independent.

In the past, one challenge was lack of a controlled environment that emulates an economy and its market where both humans and artificial agents can participate at the same time[6]. The struggle and obstacles have changed recently in experimental finance, with the introduction of platforms like O-tree (Chen, Schonger, & Wickens, 2016) and Flex-E-Markets[7]. Moreover, the financial industry have also proposed new interactive platforms for multi-agent competition and collaboration in financial decision-making tasks (Amrouni et al., 2021). Admittedly, the tasks and frameworks developed by the industry are designed for AI competitions, but extension to inclusion of human agents should be relatively straightforward.

[5]See Chapter 2 for the definition of an intelligent agent.
[6]In other fields like computer science, simulations and experiments can be conducted in community-accepted, controlled environments. For instance, the GLUE benchmark for natural language tasks (A. Wang et al., 2018), or the OpenAI gym framework for control problems (Brockman et al., 2016).
[7]https://adhocmarkets.com/.

With the help of those platforms, researchers are able to study human-robot interactions in a market setting at a granular level. Some work have already begun; for instance, Asparouhova et al. (2020) and Aldrich and López Vargas (2020) have evinced that human traders utilize trading robots with pre-determined algorithms when they are presented the opportunities to do so (Bao et al., 2021). However, much work is needed on this topic. Research of AI agents has already spanned to financial applications, and the social consequences of deploying AI in financial markets could potentially be significant, yet little is known on whether and how AI agents could shape the market and its participants' behavior.

Moreover, these platforms offer a route for the regulators to gain fruitful insights of newly proposed regulations on the interplay of tail risk and individual investors. For instance, there has been active recent discussions among both academics and policymakers on the optimal design of the market structure for retail investor[8]. In particular, SEC recently proposed to enhance individual investor order execution by involving different order priority, a significant overhaul of the market microstructure. Given the complexity of the market microstructure, agent-based simulations appear to be crucial in unveiling the underlying dynamics and investigating potential outcomes.

* * *

Finally, in my humble opinion, understanding intelligence is not a question of whether we should but rather when and how. In order to clinch a victory in building a triumphant AGI, the first step is to have a unified framework in which a general set of basic components and definitions are agreed upon among different disciplines. This premise not only echos quests by Sutton (2022), but also arises from my personal experience. Over the years, I have always struggled to decipher discipline-specific terminologies; it only appears to me in hindsight that many terms in psychology, control theory, economics, neuroscience and operations research mean exactly the same thing even though they are called differently (e.g., utility and value function). This unity/commonality is the key to understanding human learning and

[8](2022) SEC Proposes Rule to Enhance Competition for Individual Investor Order Execution (https://www.sec.gov/news/press-release/2022-225).

behavior, and it is also critical to building a generalized AI for different tasks, yet the terminology convention has become a tripwire for young apprentices. Thus, as a final remark to close this thesis, I advocate for a unified framework of AGI with common terminologies across disciplines.

Part VII

BIBLIOGRAPHY

8

BIBLIOGRAPHY

Abel, D. (2019). A theory of state abstraction for reinforcement learning. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 33, pp. 9876–9877). [Cited on page 31.]

Alberg, D., Shalit, H., & Yosef, R. (2008). Estimating stock market volatility using asymmetric garch models. *Applied Financial Economics*, *18*(15), 1201–1208. [Cited on page 68.]

Aldrich, E. M., & López Vargas, K. (2020). Experiments in high-frequency trading: comparing two market institutions. *Experimental Economics*, *23*(2), 322–352. [Cited on pages 118, 154, and 165.]

Alton, M. R., & Plott, C. R. (2007). Principles of continuous price determination in an experimental environment with flows of random arrivals and departures. *Available at SSRN 1083863*. [Cited on pages 117 and 118.]

Amihud, Y., & Mendelson, H. (1980). Dealership market: Market-making with inventory. *Journal of financial economics*, *8*(1), 31–53. [Cited on page 116.]

Amrouni, S., Moulin, A., & Balch, T. (2022). Ctmstou driven markets: simulated environment for regime-awareness in trading policies. *arXiv preprint arXiv:2202.00941*. [Cited on pages 2 and 156.]

Amrouni, S., Moulin, A., Vann, J., Vyetrenko, S., Balch, T., & 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* (pp. 1–9). [Cited on page 164.]

Anand, A., Tanggaard, C., & Weaver, D. G. (2009). Paying for market quality. *Journal of Financial and Quantitative Analysis*, *44*(6), 1427–1457. [Cited on

page 117.]

Andreoni, J., & Sprenger, C. (2011). *Uncertainty equivalents: Testing the limits of the independence axiom* (Tech. Rep.). National Bureau of Economic Research. [Cited on page 161.]

Ang, A., Chen, J., & Xing, Y. (2006). Downside risk. *The review of financial studies*, *19*(4), 1191–1239. [Cited on page 5.]

Arrow, K. J. (1951). An extension of the basic theorems of classical welfare economics. In *Proceedings of the second berkeley symposium on mathematical statistics and probability* (pp. 507–532). [Cited on pages 114 and 132.]

Asadi, K. (2015). Strengths, weaknesses, and combinations of model-based and model-free reinforcement learning. *Department of Computing Science University of Alberta*. [Cited on page 28.]

Asparouhova, E. N., Bossaerts, P., Rotaru, K., Wang, T., Yadav, N., & Yang, W. (2020). Humans in charge of trading robots: The first experiment. *Available at SSRN 3569435*. [Cited on pages 114, 118, 155, and 165.]

Bagehot, W. (1971). The only game in town. *Financial Analysts Journal*, *27*(2), 12–14. [Cited on page 116.]

Bao, T., Nekrasova, E., Neugebauer, T., & Riyanto, Y. E. (2021). Algorithmic trading in experimental markets with human traders: A literature survey. *Available at SSRN 3908065*. [Cited on pages 114, 155, and 165.]

Barber, B. M., Huang, X., Odean, T., & Schwarz, C. (2022). Attention-induced trading and returns: Evidence from robinhood users. *The Journal of Finance*, *77*(6), 3141–3190. [Cited on page 6.]

Barberis, N. (2013). The psychology of tail events: progress and challenges. *American Economic Review*, *103*(3), 611–16. [Cited on page 8.]

Barberis, N., Greenwood, R., Jin, L., & Shleifer, A. (2018). Extrapolation and bubbles. *Journal of Financial Economics*, *129*(2), 203–227. [Cited on page 156.]

Barnard, E. (1993). Temporal-difference methods and markov models. *IEEE Transactions on Systems, Man, and Cybernetics*, *23*(2), 357–365. [Cited on page 32.]

Barro, R. J. (2006). Rare disasters and asset markets in the twentieth century. *The*

Quarterly Journal of Economics, *121*(3), 823–866. [Cited on pages 5 and 6.]

Barth-Maron, G., Hoffman, M. W., Budden, D., Dabney, W., Horgan, D., Tb, D., ... Lillicrap, T. (2018). Distributed distributional deterministic policy gradients. *arXiv preprint arXiv:1804.08617*. [Cited on page 36.]

Behrens, T. E., Woolrich, M. W., Walton, M. E., & Rushworth, M. F. (2007). Learning the value of information in an uncertain world. *Nature neuroscience*, *10*(9), 1214–1221. [Cited on page 7.]

Beirlant, J., Goegebeur, Y., Segers, J., & Teugels, J. L. (2004). *Statistics of extremes: theory and applications* (Vol. 558). John Wiley & Sons. [Cited on page 5.]

Bellemare, M. G., Dabney, W., & Munos, R. (2017). A distributional perspective on reinforcement learning. In *International conference on machine learning* (pp. 449–458). [Cited on pages 33, 35, 37, 38, 39, 53, 61, 64, 65, and 73.]

Bellemare, M. G., Dabney, W., & Rowland, M. (2022). *Distributional reinforcement learning*. MIT Press. (http://www.distributional-rl.org) [Cited on page 22.]

Bellemare, M. G., Le Roux, N., Castro, P. S., & Moitra, S. (2019). Distributional reinforcement learning with linear function approximation. In *The 22nd international conference on artificial intelligence and statistics* (pp. 2203–2211). [Cited on page 35.]

Bellini, F., & Di Bernardino, E. (2017). Risk management with expectiles. *The European Journal of Finance*, *23*(6), 487–506. [Cited on page 44.]

Bellman, R., & Dreyfus, S. (1959). Functional approximations and dynamic programming. *Mathematical Tables and Other Aids to Computation*, 247–251. [Cited on page 31.]

Bellman, R. E. (1957). *Dynamic programming*. Princeton University Press. [Cited on page 63.]

Bengio, S., Bengio, Y., Cloutier, J., & Gescei, J. (2013). On the optimization of a synaptic learning rule. In *Optimality in biological and artificial networks?* (pp. 281–303). Routledge. [Cited on page 163.]

Bessembinder, H., Panayides, M., & Venkataraman, K. (2009). Hidden liquidity: an analysis of order exposure strategies in electronic stock markets. *Journal*

of Financial Economics, *94*(3), 361–383. [Cited on page 117.]

Bianchi, D., Büchner, M., & Tamoni, A. (2021). Bond risk premiums with machine learning. *The Review of Financial Studies*, *34*(2), 1046–1089. [Cited on page 2.]

Bishop, C. M. (2006). Pattern recognition. *Machine learning*, *128*(9). [Cited on page 14.]

Bogle, J. C. (2016). The index mutual fund: 40 years of growth, change, and challenge. *Financial Analysts Journal*, *72*(1), 9–13. [Cited on page 69.]

Bollerslev, T., et al. (1987). A conditionally heteroskedastic time series model for speculative prices and rates of return. *Review of economics and statistics*, *69*(3), 542–547. [Cited on page 68.]

Bordalo, P., Gennaioli, N., Ma, Y., & Shleifer, A. (2020). Overreaction in macroeconomic expectations. *American Economic Review*, *110*(9), 2748–82. [Cited on page 89.]

Bornstein, A. M., Khaw, M. W., Shohamy, D., & Daw, N. D. (2017). Reminders of past choices bias decisions for reward in humans. *Nature Communications*, *8*(1), 1–9. [Cited on page 8.]

Bossaerts, P. (2005). *The paradox of asset pricing*. Princeton University Press. [Cited on pages 20 and 59.]

Bossaerts, P. (2018). Formalizing the function of anterior insula in rapid adaptation. *Frontiers in Integrative Neuroscience*, *12*, 61. [Cited on pages 9 and 163.]

Bossaerts, P. (2022). Judgment and decision-making under uncertainty. *[Unpublished manuscript]*. [Cited on pages 8, 9, and 162.]

Bossaerts, P., Huang, S., & Yadav, N. (2020). Exploiting distributional temporal difference learning to deal with tail risk. *Risks*, *8*(4), 113. [Cited on page 106.]

Bossaerts, P., & Murawski, C. (2017). Computational complexity and human decision-making. *Trends in Cognitive Sciences*, *21*(12), 917–929. [Cited on page 7.]

Bossaerts, P., Yadav, N., & Murawski, C. (2019). Uncertainty and computational complexity. *Philosophical Transactions of the Royal Society B*, *374*(1766),

20180138. [Cited on page 7.]

Botvinick, M., Ritter, S., Wang, J. X., Kurth-Nelson, Z., Blundell, C., & Hassabis, D. (2019). Reinforcement learning, fast and slow. *Trends in cognitive sciences*, *23*(5), 408–422. [Cited on page 106.]

Botvinick, M., Wang, J. X., Dabney, W., Miller, K. J., & Kurth-Nelson, Z. (2020). Deep reinforcement learning and its neuroscientific implications. *Neuron*. [Cited on page 30.]

Botvinick, M., & Weinstein, A. (2014). Model-based hierarchical reinforcement learning and human action control. *Philosophical Transactions of the Royal Society B: Biological Sciences*, *369*(1655), 20130480. [Cited on page 106.]

Bouchaud, J.-P., Bonart, J., Donier, J., & Gould, M. (2018). *Trades, quotes and prices: financial markets under the microscope*. Cambridge University Press. [Cited on page 155.]

Brockman, G., Cheung, V., Pettersson, L., Schneider, J., Schulman, J., Tang, J., & Zaremba, W. (2016). Openai gym. *arXiv preprint arXiv:1606.01540*. [Cited on page 164.]

Brogaard, J., & Garriott, C. (2019). High-frequency trading competition. *Journal of Financial and Quantitative Analysis*, *54*(4), 1469–1497. [Cited on page 154.]

Brogaard, J., Hendershott, T., & Riordan, R. (2014). High-frequency trading and price discovery. *The Review of Financial Studies*, *27*(8), 2267–2306. [Cited on page 117.]

Brown, H., Friston, K., & Bestmann, S. (2011). Active inference, attention, and motor preparation. *Frontiers in psychology*, *2*, 218. [Cited on pages 20 and 162.]

Brown, N., & Sandholm, T. (2019). Superhuman ai for multiplayer poker. *Science*, *365*(6456), 885–890. [Cited on pages 1, 2, and 55.]

Brunnermeier, M. K., & Pedersen, L. H. (2009). Market liquidity and funding liquidity. *The review of financial studies*, *22*(6), 2201–2238. [Cited on pages 133 and 143.]

Bush, R. R., & Mosteller, F. (1951). A mathematical model for simple learning. *Psychological review*, *58*(5), 313. [Cited on page 19.]

Bush, R. R., & Mosteller, F. (1955). Stochastic models for learning. [Cited on page 19.]

Cambridge english dictionary. (2022). Cambridge University Press. Retrieved from https://dictionary.cambridge.org/dictionary/english/intelligence [Cited on page 17.]

Camerer, C. F. (2011). *Behavioral game theory: Experiments in strategic interaction*. Princeton university press. [Cited on pages 17, 18, and 19.]

Caruna, R. (1993). Multitask learning: A knowledge-based source of inductive bias. In *Machine learning: Proceedings of the tenth international conference* (pp. 41–48). [Cited on page 163.]

Casella, G., & Berger, R. L. (2021). *Statistical inference*. Cengage Learning. [Cited on pages 6, 9, 48, 66, 81, 85, 94, 95, and 96.]

Chakraborti, A., Toke, I. M., Patriarca, M., & Abergel, F. (2011). Econophysics review: Ii. agent-based models. *Quantitative Finance*, *11*(7), 1013–1041. [Cited on page 18.]

Chapkovski, P., Khapko, M., & Zoican, M. (2021). Does gamified trading stimulate risk taking? *Swedish House of Finance Research Paper*(21-25). [Cited on page 6.]

Chapman, D. G., Robbins, H., et al. (1951). Minimum variance estimation without regularity assumptions. *Annals of Mathematical Statistics*, *22*(4), 581–586. [Cited on pages 86 and 106.]

Charpentier, A., Elie, R., & Remlinger, C. (2021). Reinforcement learning in economics and finance. *Computational Economics*, 1–38. [Cited on page 21.]

Chen, D. L., Schonger, M., & Wickens, C. (2016). otree—an open-source platform for laboratory, online, and field experiments. *Journal of Behavioral and Experimental Finance*, *9*, 88–97. [Cited on page 164.]

Choi, Y., Lee, K., & Oh, S. (2019). Distributional deep reinforcement learning with a mixture of gaussians. In *2019 international conference on robotics and automation (icra)* (pp. 9791–9797). [Cited on page 36.]

Cichy, R. M., & Kaiser, D. (2019). Deep neural networks as scientific models. *Trends in cognitive sciences*, *23*(4), 305–317. [Cited on pages 15 and 57.]

Cimellaro, G. P., Noori, A. Z., Kammouh, O., Terzic, V., & Mahin, S. A. (2022). Resilience of critical structures, infrastructures and communities. *arXiv preprint arXiv:2202.09567*. [Cited on page 157.]

Cirillo, P., & Taleb, N. N. (2016). On the statistical properties and tail risk of violent conflicts. *Physica A: Statistical Mechanics and its Applications*, *452*, 29–45. [Cited on page 5.]

Cirillo, P., & Taleb, N. N. (2020). Tail risk of contagious diseases. *Nature Physics*, *16*(6), 606–613. [Cited on page 5.]

Clauset, A., Shalizi, C. R., & Newman, M. E. (2009). Power-law distributions in empirical data. *SIAM review*, *51*(4), 661–703. [Cited on pages 5, 115, 135, 137, 149, and 150.]

Clauset, A., & Woodard, R. (2013). Estimating the historical and future probabilities of large terrorist events. *The Annals of Applied Statistics*, 1838–1865. [Cited on page 5.]

Cliff, D. (1997). Minimal-intelligence agents for bargaining behaviors in market-based environments. *Hewlett-Packard Labs Technical Reports*. [Cited on pages 114, 115, 145, 153, and 200.]

Cohen, K. J., Maier, S. R., & Whitcomb, D. (1986). *The microstructure of securities markets*. Prentice Hall. [Cited on page 116.]

Cohen, M., & Jaffray, J.-Y. (1988). Certainty effect versus probability distortion: An experimental analysis of decision making under risk. *Journal of Experimental Psychology: Human Perception and Performance*, *14*(4), 554. [Cited on page 161.]

Cohen, N., Balch, T., & Veloso, M. (2020). Trading via image classification. In *Proceedings of the first acm international conference on ai in finance* (pp. 1–6). [Cited on page 3.]

Collins, A. G., & Shenhav, A. (2021). Advances in modeling learning and decision-making in neuroscience. *Neuropsychopharmacology*, 1–15. [Cited on pages 24, 31, and 162.]

Comerton-Forde, C., Hendershott, T., Jones, C. M., Moulton, P. C., & Seasholes, M. S. (2010). Time variation in liquidity: The role of market-maker inventories and revenues. *The journal of finance*, *65*(1), 295–331. [Cited on

pages 116 and 117.]

Corhay, A., & Rad, A. T. (1994). Statistical properties of daily returns: Evidence from european stock markets. *Journal of Business Finance & Accounting*, *21*(2), 271–282. [Cited on page 52.]

Cramér, H. (2016). *Mathematical methods of statistics (pms-9), volume 9*. Princeton university press. [Cited on pages 85 and 106.]

Curto, J. D., Pinto, J. C., & Tavares, G. N. (2009). Modeling stock markets' volatility using garch models with normal, student's t and stable paretian distributions. *Statistical Papers*, *50*(2), 311. [Cited on page 56.]

Cvitanić, J., Plott, C., & Tseng, C.-Y. (2015). Markets with random lifetimes and private values: mean reversion and option to trade. *Decisions in Economics and Finance*, *38*(1), 1–19. [Cited on pages 117 and 118.]

Dabney, W., Kurth-Nelson, Z., Uchida, N., Starkweather, C. K., Hassabis, D., Munos, R., & Botvinick, M. (2020). A distributional code for value in dopamine-based reinforcement learning. *Nature*, *577*(7792), 671–675. [Cited on pages 35 and 106.]

Dabney, W., Ostrovski, G., Silver, D., & Munos, R. (2018). Implicit quantile networks for distributional reinforcement learning. In *Proceedings of the international conference on machine learning*. [Cited on pages 35, 40, 43, 53, and 73.]

Dabney, W., Rowland, M., Bellemare, M. G., & Munos, R. (2018). Distributional reinforcement learning with quantile regression. In *Thirty-second aaai conference on artificial intelligence*. [Cited on pages 35, 46, 67, and 106.]

d'Acremont, M., & Bossaerts, P. (2016). Neural mechanisms behind identification of leptokurtic noise and adaptive behavioral response. *Cerebral Cortex*, *26*(4), 1818–1830. [Cited on pages 4, 5, 8, 9, and 56.]

d'Acremont, M., Schultz, W., & Bossaerts, P. (2013). The human brain encodes event frequencies while forming subjective beliefs. *Journal of Neuroscience*, *33*(26), 10887–10897. [Cited on pages 87 and 163.]

Dai, A. M., & Le, Q. V. (2015). Semi-supervised sequence learning. *Advances in neural information processing systems*, *28*. [Cited on page 163.]

Daouia, A., Girard, S., & Stupfler, G. (2018). Estimation of tail risk based on extreme expectiles. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, *80*(2), 263–292. [Cited on page 44.]

Darmois, G. (1945). Sur les limites de la dispersion de certaines estimations. *Revue de l'Institut International de Statistique*, *13*, 9-15. [Cited on page 85.]

da Silva, C. F., & Hare, T. A. (2020). Humans primarily use model-based inference in the two-stage task. *Nature Human Behaviour*, *4*(10), 1053–1066. [Cited on page 107.]

da Silveira, R. A., Sung, Y., & Woodford, M. (2020). *Optimally imprecise memory and biased forecasts* (Tech. Rep.). National Bureau of Economic Research. [Cited on pages 89 and 90.]

Daw, N. D. (2012). Model-based reinforcement learning as cognitive search: neurocomputational theories. *Cognitive search: Evolution, algorithms and the brain*, 195–208. [Cited on page 28.]

Daw, N. D., Gershman, S. J., Seymour, B., Dayan, P., & Dolan, R. J. (2011). Model-based influences on humans' choices and striatal prediction errors. *Neuron*, *69*(6), 1204–1215. [Cited on pages 106 and 161.]

Dayan, P., & Berridge, K. C. (2014). Model-based and model-free pavlovian reward learning: revaluation, revision, and revelation. *Cognitive, Affective, & Behavioral Neuroscience*, *14*(2), 473–492. [Cited on page 28.]

Dayan, P., & Niv, Y. (2008). Reinforcement learning: the good, the bad and the ugly. *Current opinion in neurobiology*, *18*(2), 185–196. [Cited on page 22.]

Dayan, P., & Yu, A. J. (2002). Expected and unexpected uncertainty: Ach and ne in the neocortex. *Advances in neural information processing systems*, *15*. [Cited on pages 7 and 8.]

Debreu, G. (1951). The coefficient of resource utilization. *Econometrica: Journal of the Econometric Society*, 273–292. [Cited on pages 114 and 132.]

Debreu, G. (1954). Valuation equilibrium and pareto optimum. *Proceedings of the National Academy of Sciences of the United States of America*, *40*(7), 588. [Cited on pages 114 and 132.]

De Haan, L., & Ferreira, A. (2006). *Extreme value theory: an introduction* (Vol. 21). Springer. [Cited on page 5.]

De Luca, M., Szostek, C., Cartlidge, J., & Cliff, D. (2011). Studies of interaction between human traders and algorithmic trading systems. *Foresight Project*. [Cited on page 115.]

De Prado, M. L. (2018). *Advances in financial machine learning*. John Wiley & Sons. [Cited on page 16.]

De Rossi, G., & Harvey, A. (2009). Quantiles, expectiles and splines. *Journal of Econometrics*, *152*(2), 179–185. [Cited on page 44.]

Devlin, J., Chang, M. W., Lee, K., & Toutanova, K. (2019). BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 conference of the north american chapter of the association for computational linguistics* (pp. 4171–4186). Association for Computational Linguistics. doi: 10.18653/v1/n19-1423 [Cited on page 163.]

Dewanto, V., Dunn, G., Eshragh, A., Gallagher, M., & Roosta, F. (2020). Average-reward model-free reinforcement learning: a systematic review and literature mapping. *arXiv preprint arXiv:2010.08920*. [Cited on page 161.]

Dixon, M. F., Halperin, I., & Bilokon, P. (2020). *Machine learning in finance* (Vol. 1406). Springer. [Cited on page 21.]

Drummond, N., & Niv, Y. (2020). Model-based decision making and model-free learning. *Current Biology*, *30*(15), R860–R865. [Cited on page 28.]

Duffie, D., Gârleanu, N., & Pedersen, L. H. (2005). Over-the-counter markets. *Econometrica*, *73*(6), 1815–1847. [Cited on page 116.]

Duffie, D., & Pan, J. (1997). An overview of value at risk. *Journal of derivatives*, *4*(3), 7–49. [Cited on page 5.]

Duffy, J. (2006). Agent-based models and human subject experiments. *Handbook of computational economics*, *2*, 949–1011. [Cited on page 114.]

Easley, D., & O'Hara, M. (2003). Microstructure and asset pricing. *Handbook of the Economics of Finance*, *1*, 1021–1051. [Cited on pages 153 and 154.]

Engel, Y., Mannor, S., & Meir, R. (2005). Reinforcement learning with gaussian processes. In *Proceedings of the 22nd international conference on machine learning* (pp. 201–208). [Cited on pages 24 and 36.]

Eysenck, M. (2014). *Fundamentals of psychology*. Psychology Press. [Cited on pages 17 and 18.]

Farmer, J. D., & Lillo, F. (2004). On the origin of power-law tails in price fluctuations. *Quantitative Finance*, *4*(1), C7. [Cited on pages 9, 115, and 155.]

Farrell, M. H., Liang, T., & Misra, S. (2021). Deep neural networks for estimation and inference. *Econometrica*, *89*(1), 181–213. [Cited on page 3.]

Finn, C. B. (2018). *Learning to learn with gradients*. University of California, Berkeley. [Cited on pages 1 and 163.]

Fischer, T. G. (2018). *Reinforcement learning in financial markets-a survey* (Tech. Rep.). FAU Discussion Papers in Economics. [Cited on page 21.]

Franses, P. H., Van Der Leij, M., & Paap, R. (2007). A simple test for garch against a stochastic volatility model. *Journal of Financial Econometrics*, *6*(3), 291–306. [Cited on page 68.]

Fréchet, M. (1943). Sur l'extension de certaines évaluations statistiques au case de petits échantillonsles limites de la dispersion de certaines estimations. *Revue de l'Institut International de Statistique*, *11*, 182-205. [Cited on page 85.]

Friedman, D. (1998). Evolutionary economics goes mainstream: A review of the theory of learning in games. *Journal of Evolutionary Economics*, *8*(4), 423–432. [Cited on page 19.]

Friedman, J. A. (2015). Using power laws to estimate conflict size. *Journal of Conflict Resolution*, *59*(7), 1216–1241. [Cited on page 5.]

Friston, K., FitzGerald, T., Rigoli, F., Schwartenbeck, P., Pezzulo, G., et al. (2016). Active inference and learning. *Neuroscience & Biobehavioral Reviews*, *68*, 862–879. [Cited on pages 20 and 162.]

Fudenberg, D., Drew, F., Levine, D. K., & Levine, D. K. (1998). *The theory of learning in games* (Vol. 2). MIT press. [Cited on page 19.]

Fudenberg, D., & Puri, I. (2022). Simplicity and probability weighting in choice under risk. In *Aea papers and proceedings* (Vol. 112, pp. 421–25). [Cited on page 161.]

Gabaix, X. (2012). Variable rare disasters: An exactly solved framework for ten puzzles in macro-finance. *The Quarterly journal of economics*, *127*(2), 645–700. [Cited on page 6.]

Gabaix, X., Gopikrishnan, P., Plerou, V., & Stanley, H. E. (2003). A theory of power-law distributions in financial market fluctuations. *Nature*, *423*(6937), 267–270. [Cited on pages 5 and 115.]

Ganesh, S., Vadori, N., Xu, M., Zheng, H., Reddy, P., & Veloso, M. (2019). Reinforcement learning for market making in a multi-agent dealer market. *arXiv preprint arXiv:1911.05892*. [Cited on pages 2 and 156.]

Garman, M. B. (1976). Market microstructure. *Journal of financial Economics*, *3*(3), 257–275. [Cited on page 116.]

Garnier, S., Gautrais, J., & Theraulaz, G. (2007). The biological principles of swarm intelligence. *Swarm intelligence*, *1*(1), 3–31. [Cited on pages 21, 115, and 153.]

Ghavamzadeh, M., & Engel, Y. (2007). Bayesian actor-critic algorithms. In *Proceedings of the 24th international conference on machine learning* (pp. 297–304). [Cited on page 36.]

Glasserman, P. (2013). *Monte carlo methods in financial engineering* (Vol. 53). Springer Science & Business Media. [Cited on page 55.]

Glimcher, P. W. (2011). Understanding dopamine and reinforcement learning: the dopamine reward prediction error hypothesis. *Proceedings of the National Academy of Sciences*, *108*(Supplement 3), 15647–15654. [Cited on page 22.]

Glivenko, V. (1933). Sulla determinazione empirica delle leggi di probabilita. *Gion. Ist. Ital. Attauri.*, *4*, 92–99. [Cited on page 136.]

Glosten, L. R., & Milgrom, P. R. (1985). Bid, ask and transaction prices in a specialist market with heterogeneously informed traders. *Journal of financial economics*, *14*(1), 71–100. [Cited on pages 116, 126, and 154.]

Gode, D. K., & Sunder, S. (1993). Allocative efficiency of markets with zero-intelligence traders: Market as a partial substitute for individual rationality. *Journal of political economy*, *101*(1), 119–137. [Cited on pages 9, 21, 115, 132, and 153.]

Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep learning*. MIT press. [Cited on page 14.]

Gu, S., Kelly, B., & Xiu, D. (2020). Empirical asset pricing via machine learning. *The Review of Financial Studies*, *33*(5), 2223–2273. [Cited on pages 2, 21,

and 156.]

Hammersley, J. M. (1950). On estimating restricted parameters. *Journal of the Royal Statistical Society. Series B (Methodological)*, *12*(2), 192–240. [Cited on page 86.]

Harris, L., Kahn, C. M., McDonald, R. L., & Spatt, C. S. (2021). The role of pilot studies in financial regulation. *Available at SSRN*. [Cited on page 156.]

Hasbrouck, J., & Seppi, D. J. (2001). Common factors in prices, order flows, and liquidity. *Journal of financial Economics*, *59*(3), 383–411. [Cited on page 116.]

Hendershott, T., Jones, C. M., & Menkveld, A. J. (2011). Does algorithmic trading improve liquidity? *The Journal of finance*, *66*(1), 1–33. [Cited on page 117.]

Ho, T., & Stoll, H. R. (1981). Optimal dealer pricing under transactions and return uncertainty. *Journal of Financial economics*, *9*(1), 47–73. [Cited on page 116.]

Jovanovic, B., & Menkveld, A. J. (2016). Middlemen in limit order markets. *Available at SSRN 1624329*. [Cited on page 154.]

Jurczenko, E., & Maillet, B. (2012). The four-moment capital asset pricing model: between asset pricing and asset allocation. *Multi-moment Asset Allocation and Pricing Models*, 113–163. [Cited on page 55.]

Kai-Ineman, D., & Tversky, A. (1979). Prospect theory: An analysis of decision under risk. *Econometrica*, *47*(2), 363–391. [Cited on page 160.]

Kalda, A., Loos, B., Previtero, A., & Hackethal, A. (2021). *Smart (phone) investing? a within investor-time analysis of new technologies and trading behavior*. (Tech. Rep.). National Bureau of Economic Research. [Cited on page 6.]

Kapturowski, S., Campos, V., Jiang, R., Rakićević, N., Hasselt, H. v., Blundell, C., & Badia, A. P. (2022). Human-level atari 200x faster. *arXiv preprint arXiv:2209.07550*. [Cited on page 2.]

Karpe, M., Fang, J., Ma, Z., & 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* (pp. 1–7). [Cited on page 2.]

Kelly, B., & Jiang, H. (2014). Tail risk and asset prices. *The Review of Financial Studies*, *27*(10), 2841–2871. [Cited on pages 5 and 155.]

Kirilenko, A., Kyle, A. S., Samadi, M., & Tuzun, T. (2017). The flash crash: High-frequency trading in an electronic market. *The Journal of Finance*, *72*(3), 967–998. [Cited on page 138.]

Kneib, T. (2013). Beyond mean regression. *Statistical Modelling*, *13*(4), 275–303. [Cited on pages 44 and 48.]

Koenker, R., & Hallock, K. F. (2001). Quantile regression. *Journal of economic perspectives*, *15*(4), 143–156. [Cited on page 41.]

Koijen, R. S., Richmond, R. J., & Yogo, M. (2020). *Which investors matter for equity valuations and expected returns?* (Tech. Rep.). National Bureau of Economic Research. [Cited on page 18.]

Kyle, A. S. (1985). Continuous auctions and insider trading. *Econometrica: Journal of the Econometric Society*, 1315–1335. [Cited on pages 116, 126, and 154.]

Lagos, R., & Rocheteau, G. (2009). Liquidity in asset markets with search frictions. *Econometrica*, *77*(2), 403–426. [Cited on page 116.]

Lee, S. Y., Sungik, C., & Chung, S.-Y. (2019). Sample-efficient deep reinforcement learning via episodic backward update. *Advances in Neural Information Processing Systems*, *32*. [Cited on page 1.]

Leippold, M., Wang, Q., & Zhou, W. (2022). Machine learning in the chinese stock market. *Journal of Financial Economics*, *145*(2), 64–82. [Cited on pages 2, 3, and 156.]

Liu, C., & Rubin, D. B. (1995). Ml estimation of the t distribution using em and its extensions, ecm and ecme. *Statistica Sinica*, 19–39. [Cited on pages 71, 94, and 101.]

Lowet, A. S., Zheng, Q., Matias, S., Drugowitsch, J., & Uchida, N. (2020). Distributional reinforcement learning in the brain. *Trends in Neurosciences*. [Cited on page 35.]

Ludvig, E. A., Bellemare, M. G., & Pearson, K. G. (2011). A primer on reinforcement learning in the brain: Psychological, computational, and neural perspectives. *Computational neuroscience for advancing artificial*

intelligence: Models, methods and applications, 111–144. [Cited on pages 15 and 57.]

Lyle, C., Bellemare, M. G., & Castro, P. S. (2019). A comparative analysis of expected and distributional reinforcement learning. In *Proceedings of the aaai conference on artificial intelligence* (Vol. 33, pp. 4504–4511). [Cited on pages 50 and 53.]

Madan, D. B. (2017). Efficient estimation of expected stock price returns. *Finance Research Letters*, *23*, 31–38. [Cited on page 54.]

Mahadevan, S. (1996). Average reward reinforcement learning: Foundations, algorithms, and empirical results. *Machine learning*, *22*(1), 159–195. [Cited on page 161.]

Mandelbrot, B. (1963). The variation of certain speculative prices. *The Journal of Business*, *36*(4), 394-419. [Cited on pages 4 and 5.]

March, C. (2019). The behavioral economics of artificial intelligence: Lessons from experiments with computer players.
[Cited on pages 114 and 155.]

Markowitz, H. (1952). Portfolio selection. *The Journal of Finance*, *7*(1), 77-91. doi: 10.2307/2975974 [Cited on page 5.]

Mas-Colell, A., Whinston, M. D., & Green, J. R. (1995). *Microeconomic theory* (Vol. 1). Oxford university press New York. [Cited on page 132.]

McCarthy, J. (2007). *What is artificial intelligence.* Retrieved from http://www-formal.stanford.edu/jmc/whatisai/ [Cited on pages 14 and 17.]

Menkveld, A. J., & Zoican, M. A. (2017). Need for speed? exchange latency and liquidity. *The Review of Financial Studies*, *30*(4), 1188–1228. [Cited on pages 117 and 154.]

Mittnik, S., Paolella, M. S., & Rachev, S. T. (1998). Unconditional and conditional distributional models for the nikkei index. *Asia-Pacific Financial Markets*, *5*(2), 99. [Cited on page 68.]

Mnih, V., Kavukcuoglu, K., Silver, D., Rusu, A. A., Veness, J., Bellemare, M. G., ... others (2015). Human-level control through deep reinforcement learning. *nature*, *518*(7540), 529–533. [Cited on pages 1, 2, 20, 33, 34, and 55.]

Modigliani, F., & Miller, M. H. (1958). The cost of capital, corporation finance and the theory of investment. *The American economic review*, *48*(3), 261–297. [Cited on page 23.]

Moerland, T. M., Broekens, J., & Jonker, C. M. (2020). Model-based reinforcement learning: A survey. *arXiv preprint arXiv:2006.16712*. [Cited on page 28.]

Moravčík, M., Schmid, M., Burch, N., Lisỳ, V., Morrill, D., Bard, N., ... Bowling, M. (2017). Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. *Science*, *356*(6337), 508–513. [Cited on page 1.]

Morimura, T., Sugiyama, M., Kashima, H., Hachiya, H., & Tanaka, T. (2010). Nonparametric return distribution approximation for reinforcement learning. In *Icml*. [Cited on page 33.]

Morimura, T., Sugiyama, M., Kashima, H., Hachiya, H., & Tanaka, T. (2012). Parametric return density estimation for reinforcement learning. *arXiv preprint arXiv:1203.3497*. [Cited on pages 33 and 36.]

Naik, D. K., & Mammone, R. J. (1992). Meta-neural networks that learn by learning. In *[proceedings 1992] ijcnn international joint conference on neural networks* (Vol. 1, pp. 437–442). [Cited on page 163.]

Nassar, M. R., Wilson, R. C., Heasly, B., & Gold, J. I. (2010). An approximately bayesian delta-rule model explains the dynamics of belief updating in a changing environment. *Journal of Neuroscience*, *30*(37), 12366–12378. [Cited on page 90.]

Navarro, D. J., Tran, P., & Baz, N. (2018). Aversion to option loss in a restless bandit task. *Computational Brain & Behavior*, *1*(2), 151–164. [Cited on page 8.]

Nesterov, Y. (1998). Introductory lectures on convex programming volume i: Basic course. *Lecture notes*, *3*(4), 5. [Cited on page 26.]

Newey, W. K., & Powell, J. L. (1987). Asymmetric least squares estimation and testing. *Econometrica: Journal of the Econometric Society*, 819–847. [Cited on page 43.]

Nilsson, N. J. (2009). *The quest for artificial intelligence*. Cambridge University Press. [Cited on pages 14, 15, and 17.]

Niv, Y. (2009). Reinforcement learning in the brain. *Journal of Mathematical Psychology*, *53*(3), 139–154. [Cited on pages 22 and 28.]

Nowak, P., & Romaniuk, M. (2013). A fuzzy approach to option pricing in a levy process setting. *International Journal of Applied Mathematics and Computer Science*, *23*(3), 613–622. [Cited on page 55.]

Nursimulu, A., & Bossaerts, P. (2014). Excessive volatility is also a feature of individual level forecasts. *Journal of Behavioral Finance*, *15*(1), 16–29. [Cited on page 89.]

O'Hara, M. (2003). Presidential address: Liquidity and price discovery. *The journal of Finance*, *58*(4), 1335–1354. [Cited on page 153.]

Orlowski, L. T. (2012). Financial crisis and extreme market risks: Evidence from europe. *Review of Financial Economics*, *21*(3), 120–130. [Cited on pages 5 and 155.]

Palmer, R. G., Arthur, W. B., Holland, J. H., LeBaron, B., & Tayler, P. (1994). Artificial economic life: a simple model of a stockmarket. *Physica D: Nonlinear Phenomena*, *75*(1-3), 264–274. [Cited on page 116.]

Parr, T., Pezzulo, G., & Friston, K. J. (2022). *Active inference: the free energy principle in mind, brain, and behavior*. MIT Press. [Cited on page 162.]

Payzan-LeNestour, E., & Bossaerts, P. (2011). Risk, unexpected uncertainty, and estimation uncertainty: Bayesian learning in unstable settings. *PLoS computational biology*, *7*(1), e1001048. [Cited on page 7.]

Payzan-LeNestour, E., Dunne, S., Bossaerts, P., & O'Doherty, J. P. (2013). The neural representation of unexpected uncertainty during value-based decision making. *Neuron*, *79*(1), 191–201. [Cited on pages 7 and 8.]

Pisarenko, V., & Rodkin, M. (2010). *Heavy-tailed distributions in disaster analysis* (Vol. 30). Springer Science & Business Media. [Cited on page 5.]

Poggio, T., & Serre, T. (2013). Models of visual cortex. *Scholarpedia*, *8*(4), 3516. [Cited on pages 15 and 57.]

Prashanth, L., Jie, C., Fu, M., Marcus, S., & Szepesvári, C. (2016). Cumulative prospect theory meets reinforcement learning: Prediction and control. In *International conference on machine learning* (pp. 1406–1415). [Cited on page 161.]

Puri, I. (2022). Simplicity and risk. *Available at SSRN 3253494*. [Cited on page 161.]

Radford, A., Narasimhan, K., Salimans, T., Sutskever, I., et al. (2018). Improving language understanding by generative pre-training. OpenAI. Retrieved from https://openai.com/blog/language-unsupervised/ [Cited on page 163.]

Rao, C. R. (1992). Information and the accuracy attainable in the estimation of statistical parameters. In *Breakthroughs in statistics* (pp. 235–247). Springer. [Cited on page 85.]

Reed, S., Zolna, K., Parisotto, E., Colmenarejo, S. G., Novikov, A., Barth-Maron, G., ... others (2022). A generalist agent. *arXiv preprint arXiv:2205.06175*. [Cited on page 163.]

Rietz, T. A. (1988). The equity risk premium a solution. *Journal of monetary Economics*, *22*(1), 117–131. [Cited on page 6.]

Rockafellar, R. T., & Uryasev, S. (2000). Optimization of conditional value-at-risk. *Journal of risk*, *2*, 21–42. [Cited on pages 81 and 115.]

Rowland, M., Bellemare, M., Dabney, W., Munos, R., & Teh, Y. W. (2018). An analysis of categorical distributional reinforcement learning. In *International conference on artificial intelligence and statistics* (pp. 29–37). [Cited on pages 53 and 73.]

Rowland, M., Dadashi, R., Kumar, S., Munos, R., Bellemare, M. G., & Dabney, W. (2019). Statistics and samples in distributional reinforcement learning. In *International conference on machine learning* (pp. 5528–5536). [Cited on pages 35, 43, 44, 45, 46, 50, 53, 64, 81, and 82.]

Roy, A. D. (1952). Safety first and the holding of assets. *Econometrica: Journal of the econometric society*, 431–449. [Cited on page 5.]

Russell, S., & Norvig, P. (2020). *Artificial intelligence: A modern approach* (4th ed.). Pearson. [Cited on page 14.]

Russell, S. J. (2010). *Artificial intelligence a modern approach*. Pearson Education, Inc. [Cited on pages 17 and 18.]

Sajid, N., Ball, P. J., Parr, T., & Friston, K. J. (2021). Active inference: demystified and compared. *Neural computation*, *33*(3), 674–712. [Cited on pages 20, 162, and 163.]

Samuel, A. L. (1967). Some studies in machine learning using the game of checkers. ii—recent progress. *IBM Journal of research and development*, *11*(6), 601–617. [Cited on page 31.]

Samuel, A. L. (2000). Some studies in machine learning using the game of checkers. *IBM Journal of research and development*, *44*(1.2), 206–226. [Cited on page 31.]

Sato, M., Kimura, H., & Kobayashi, S. (2001). Td algorithm for the variance of return and mean-variance reinforcement learning. *Transactions of the Japanese Society for Artificial Intelligence*, *16*(3), 353–362. [Cited on page 36.]

Savage, L. J. (1972). *The foundations of statistics*. Courier Corporation. [Cited on page 58.]

Scherer, M., Rachev, S. T., Kim, Y. S., & Fabozzi, F. J. (2012). Approximation of skewed and leptokurtic return distributions. *Applied Financial Economics*, *22*(16), 1305–1316. [Cited on page 55.]

Schervish, M. J. (2012). *Theory of statistics*. Springer Science & Business Media. [Cited on pages 66, 81, 82, 85, and 96.]

Schmidhuber, J. (2007). Gödel machines: Fully self-referential optimal universal self-improvers. In *Artificial general intelligence* (pp. 199–226). Springer. [Cited on page 163.]

Schultz, W., Dayan, P., & Montague, P. R. (1997). A neural substrate of prediction and reward. *Science*, *275*(5306), 1593–1599. [Cited on pages 59 and 106.]

Shalizi, C. R. (2022). *Advanced statistics from an elementary point of view*. Unpublished book. Retrieved from https://www.stat.cmu.edu/~cshalizi/ADAfaEPoV/ [Cited on page 38.]

Shen, Y., Tobia, M. J., Sommer, T., & Obermayer, K. (2014). Risk-sensitive reinforcement learning. *Neural computation*, *26*(7), 1298–1328. [Cited on page 161.]

Shoham, Y., & Leyton-Brown, K. (2008). *Multiagent systems: Algorithmic, game-theoretic, and logical foundations*. Cambridge University Press. [Cited on page 164.]

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., Van Den Driessche, G., ... Hassabis, D. (2016). Mastering the game of go with deep neural networks and tree search. *nature*, *529*(7587), 484–489. [Cited on pages 1, 2, 24, and 55.]

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., ... others (2017). Mastering the game of go without human knowledge. *nature*, *550*(7676), 354–359. [Cited on pages 1 and 20.]

Silver, D., Singh, S., Precup, D., & Sutton, R. S. (2021). Reward is enough. *Artificial Intelligence*, 103535. [Cited on pages 8, 17, 24, and 162.]

Simonato, J.-G. (2012). Garch processes with skewed and leptokurtic innovations: Revisiting the johnson su case. *Finance Research Letters*, *9*(4), 213–219. [Cited on pages 55 and 56.]

Sims, C. A. (1980). Macroeconomics and reality. *Econometrica: journal of the Econometric Society*, 1–48. [Cited on page 115.]

Singh, S., & Dayan, P. (1998). Analytical mean squared error curves for temporal difference learning. *Machine Learning*, *32*(1), 5–40. [Cited on page 56.]

Smith, V. L. (1962). An experimental study of competitive market behavior. *Journal of political economy*, *70*(2), 111–137. [Cited on pages 9, 21, 114, 115, and 132.]

Song, M., Bnaya, Z., & Ma, W. J. (2019). Sources of suboptimality in a minimalistic explore–exploit task. *Nature human behaviour*, *3*(4), 361–368. [Cited on page 30.]

Sun, Q., Zhou, W.-X., & Fan, J. (2020). Adaptive huber regression. *Journal of the American Statistical Association*, *115*(529), 254–265. [Cited on page 71.]

Sutton, R. S. (2020). John mccarthy's definition of intelligence. *Journal of Artificial General Intelligence*, *11*(2), 66–67. [Cited on pages 14 and 17.]

Sutton, R. S. (2022). The quest for a common model of the intelligent decision maker. *arXiv preprint arXiv:2202.13252*. [Cited on page 165.]

Sutton, R. S., & Barto, A. G. (2018). *Reinforcement learning: An introduction*. MIT press. [Cited on pages 15, 19, 22, 28, 30, 31, 32, 56, 63, 64, and 106.]

Tadelis, S. (2013). *Game theory: an introduction*. Princeton university press. [Cited on page 24.]

Taleb, N. N. (2007). *The black swan: The impact of the highly improbable* (Vol. 2). Random house. [Cited on pages 4 and 52.]

Taleb, N. N. (2020). *Statistical consequences of fat tails*. STEM Academic Press. [Cited on page 4.]

Taylor, J. W. (1999). A quantile regression approach to estimating the distribution of multiperiod returns. *The Journal of Derivatives*, *7*(1), 64–78. [Cited on page 40.]

Taylor, J. W. (2008). Estimating value at risk and expected shortfall using expectiles. *Journal of Financial Econometrics*, *6*(2), 231–252. [Cited on page 44.]

Toenger, S., Godin, T., Billet, C., Dias, F., Erkintalo, M., Genty, G., & Dudley, J. M. (2015). Emergent rogue wave structures and statistics in spontaneous modulation instability. *Scientific reports*, *5*, 10380. [Cited on page 52.]

Torrey, L., & Shavlik, J. (2010). Transfer learning. In *Handbook of research on machine learning applications and trends: algorithms, methods, and techniques* (pp. 242–264). IGI global. [Cited on page 163.]

Tucker, H. G. (1959). A generalization of the glivenko-cantelli theorem. *The Annals of Mathematical Statistics*, *30*(3), 828–830. [Cited on page 136.]

Tversky, A., & Kahneman, D. (1986). The framing of decisions and the evaluation of prospects. In *Studies in logic and the foundations of mathematics* (Vol. 114, pp. 503–520). Elsevier. [Cited on page 161.]

Vamplew, P., Smith, B. J., Källström, J., Ramos, G., Rădulescu, R., Roijers, D. M., ... others (2022). Scalar reward is not enough: A response to silver, singh, precup and sutton (2021). *Autonomous Agents and Multi-Agent Systems*, *36*(2), 1–19. [Cited on page 162.]

Vapnik, V. (2013). *The nature of statistical learning theory*. Springer science & business media. [Cited on page 48.]

Vinyals, O., Babuschkin, I., Czarnecki, W. M., Mathieu, M., Dudzik, A., Chung, J., ... others (2019). Grandmaster level in starcraft ii using multi-agent reinforcement learning. *Nature*, *575*(7782), 350–354. [Cited on pages 1 and 2.]

Vuong, Q. H. (1989). Likelihood ratio tests for model selection and non-nested hypotheses. *Econometrica: Journal of the Econometric Society*, 307–333. [Cited on page 137.]

Waltrup, L. S., Sobotka, F., Kneib, T., & Kauermann, G. (2015). Expectile and quantile regression—david and goliath? *Statistical Modelling*, *15*(5), 433–456. [Cited on pages 44 and 48.]

Wang, A., Singh, A., Michael, J., Hill, F., Levy, O., & Bowman, S. R. (2018). Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*. [Cited on page 164.]

Wang, P. (2019). On defining artificial intelligence. *Journal of Artificial General Intelligence*, *10*(2), 1–37. [Cited on page 17.]

Watkins, C. J., & Dayan, P. (1992). Q-learning. *Machine learning*, *8*(3-4), 279–292. [Cited on pages 28, 29, and 64.]

Weaver, D. (2012). Minimum obligations of market makers. *Foresight Project*. [Cited on page 116.]

Weiss, K., Khoshgoftaar, T. M., & Wang, D. (2016). A survey of transfer learning. *Journal of Big data*, *3*(1), 1–40. [Cited on page 163.]

Wong, F., & Collins, J. J. (2020). Evidence that coronavirus superspreading is fat-tailed. *Proceedings of the National Academy of Sciences*, *117*(47), 29416–29418. [Cited on pages 85 and 86.]

Yang, D., Zhao, L., Lin, Z., Qin, T., Bian, J., & Liu, T.-Y. (2019). Fully parameterized quantile function for distributional reinforcement learning. *Advances in neural information processing systems*, *32*, 6193–6202. [Cited on page 43.]

Yang, X.-S., Deb, S., Fong, S., He, X., & Zhao, Y.-X. (2016). From swarm intelligence to metaheuristics: nature-inspired optimization algorithms. *Computer*, *49*(9), 52–59. [Cited on pages 21, 115, and 153.]

Ye, W., Liu, S., Kurutach, T., Abbeel, P., & Gao, Y. (2021). Mastering atari games with limited data. *Advances in Neural Information Processing Systems*, *34*, 25476–25488. [Cited on page 1.]

Yu, Y. (2018). Towards sample efficient reinforcement learning. In *Ijcai* (pp. 5739–5743). [Cited on page 2.]

Zhou, F., Wang, J., & Feng, X. (2020). Non-crossing quantile regression for distributional reinforcement learning. *Advances in Neural Information Processing Systems*, *33*, 15909–15919. [Cited on page 48.]

Part VIII

APPENDIX

A

APPENDIX

A.1 Chapter 5

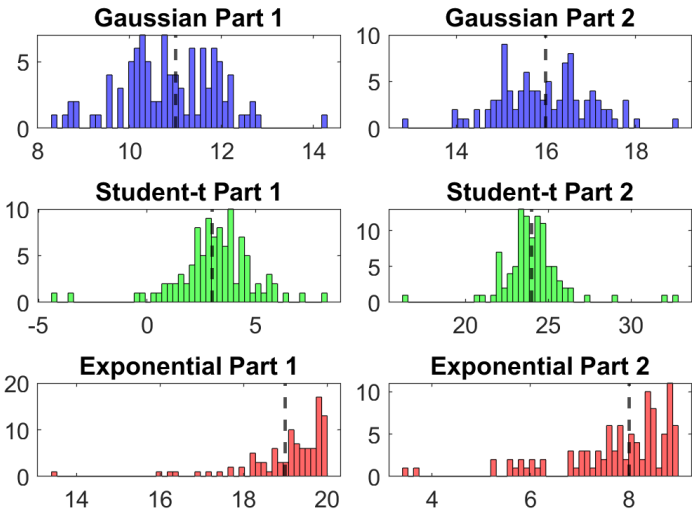


Figure A.1. Histogram of the displayed values from an example participant. True mean used in the data generating process (indicated by the black dashed vertical line): Gaussian part 1: 11.0, part 2: 16.0; Student-t part 1: 3.0, part 2: 24.0; Exponential part 1: 19.0, part 2: 8.0

	Gaussian		Student-t		Exponential	
Part	P1	P2	P1	P2	P1	P2
Mean	10.89	15.96	3.13	24.08	19.04	7.72
SD	1.05	1.01	1.79	1.91	1.02	1.14
Min	8.38	12.84	-4.38	16.31	13.44	3.41
Max	14.25	18.94	8.28	32.82	19.97	8.99
Skewness	-0.02	-0.02	-0.91	1.10	-2.42	-1.41
Kurtosis	0.18	0.28	3.87	8.19	8.49	2.06

Table A.1. Summary statistics of the displayed values in Fig A.1.

Notation		Explanation
$M_{i,t}$	Midpoint	(slider min + slider max) / 2
$P_{i,t}$	PartEst	participants' estimation
$SA_{i,t}$	sample_average	running sample average
$EE_{i,t}$	efficient_estimation	running efficient estimation
$DP_{i,t}$	DiffPredRange	participants' estimation – mid-point
$DA_{i,t}$	DiffAvgeRange	sample average – mid-point
$DE_{i,t}$	DiffEffRange	efficient estimation – mid-point
$DE_{i,t-1}$	DiffEffRangeLag	DiffEffRange 1 lag
$DE_{i,t-2}$	DiffEffRangeLag2	DiffEffRange 2 lags
$OR_{i,t}$	OrthogonalResiduals	residual values from regressing DiffEffRange on DiffAvgeRange
i	Subject	participant index: [1..44]
t	Episode	episode index: [1..20]
R	Repeat	part 1 or part 2: $\{0, 1\}$

Table A.2. A summary of notations. The abbreviations in the first column correspond to the notations used in the paper. The notations in the second column correspond the name of the variables used in the Matlab analysis code. The third column provides a short explanation.

	Gaussian	**Student-t**	**Exponential**
(Intercept)	$\beta = -0.0592$ $SE = 0.0188$ $t_{1758} = -3.15$ ***p = 0.0017***	$\beta = +0.0153$ $SE = 0.0327$ $t_{1757} = +0.47$ $p = 0.64$	$\beta = -0.0166$ $SE = 0.0271$ $t_{1579} = -0.61$ $p = 0.54$
DiffAvgeRange	$\beta = +0.724$ $SE = 0.0313$ $t_{1758} = +23.12$ ***p <0.0001***	$\beta = +0.835$ $SE = 0.0239$ $t_{1757} = +34.93$ ***p <0.0001***	$\beta = +0.711$ $SE = 0.0342$ $t_{1579} = +20.83$ ***p <0.0001***
OrthogonalResiduals		$\beta = +0.267$ $SE = 0.166$ $t_{1757} = +1.61$ ***p=0.108***	$\beta = -0.251$ $SE = 0.102$ $t_{1579} = -2.45$ ***p = 0.014***
DiffEffRangeLag			$\beta = -0.0567$ $SE = 0.0151$ $t_{1579} = -3.75$ ***p = 0.00018***
DiffEffRangeLag2			$\beta = -0.0186$ $SE = 0.0104$ $t_{1579} = -1.79$ ***p=0.07***
N_{obs}	1760	1760	1584
$adj - R^2$	0.72	0.95	0.89
BIC	452.14	2816.56	99.34
AIC	419.30	2761.83	-13.38

Table A.3. Fixed-effects coefficient (full results).

Gaussian		Type	Estimate	Lower	Upper
(Intercept)	(Intercept)	SD	0.17	0.14	0.20
DiffAvgeRange	(Intercept)	corr	0.11	-0.14	0.34
DiffAvgeRange	DiffAvgeRange	SD	0.26	0.22	0.32
Student-t					
(Intercept)	(Intercept)	SD	0.27	0.23	0.33
OrthogonalResiduals	(Intercept)	corr	0.05	-0.24	0.33
DiffAvgeRange	(Intercept)	corr	0.04	-0.20	0.28
OrthogonalResiduals	OrthogonalResiduals	SD	1.21	0.95	1.54
DiffAvgeRange	OrthogonalResiduals	corr	-0.01	-0.28	0.26
DiffAvgeRange	DiffAvgeRange	SD	0.21	0.18	0.25
Exponential					
(Intercept)	(Intercept)	SD	0.24	0.20	0.28
OrthogonalResiduals	(Intercept)	corr	0.33	NaN	NaN
DiffEffRangeLag	(Intercept)	corr	0.41	NaN	NaN
DiffEffRangeLag2	(Intercept)	corr	0.63	NaN	NaN
DiffAvgeRange	(Intercept)	corr	-0.02	NaN	NaN
OrthogonalResiduals	OrthogonalResiduals	SD	0.68	0.51	0.90
DiffEffRangeLag	OrthogonalResiduals	corr	0.49	NaN	NaN
DiffEffRangeLag2	OrthogonalResiduals	corr	0.64	NaN	NaN
DiffAvgeRange	OrthogonalResiduals	corr	-0.04	NaN	NaN
DiffEffRangeLag	DiffEffRangeLag	SD	0.11	0.08	0.14
DiffEffRangeLag2	DiffEffRangeLag	corr	0.94	NaN	NaN
DiffAvgeRange	DiffEffRangeLag	corr	0.07	-0.04	0.18
DiffEffRangeLag2	DiffEffRangeLag2	SD	0.04	0.02	0.07
DiffAvgeRange	DiffEffRangeLag2	corr	-0.12	NaN	NaN
DiffAvgeRange	DiffAvgeRange	SD	0.30	0.25	0.36

Table A.4. Random-effects correlation (full results), the lower (upper) column corresponds to the upper (lower) confidence bound at 5% level.

Number of outliers	Part 1	Part 2
Efficient	27.60 (1.02)	28.07 (1.51)
Non-Efficient	27.48 (0.84)	26.28 (0.73)

Table A.5. Total number of outliers encountered by participants in Treatment T, average across participants in each group (with SE). Outliers are defined as |displayed sample value - true mean| $>$ 1.5 SD.

Gaussian	Part 1	Part 2
Efficient	401.57 (18.17)	343.10 (9.34)
Non-Efficient	396.56 (13.52)	346.09 (11.11)

Student-t	Part 1	Part 2
Efficient	508.33 (15.59)	398.47 (13.10)
Non-Efficient	508.86 (13.24)	361.83 (8.50)

Exponential	Part 1	Part 2
Efficient	400.89 (20.79)	319.40 (15.56)
Non-Efficient	393.97 (10.92)	323.28 (7.71)

Table A.6. Total seconds spent on the sampling task, average across participants in each group (with SE).

Gaussian	Part 1	Part 2
Efficient	226.38 (8.26)	197.71 (6.14)
Non-Efficient	227.96 (8.58)	199.13 (7.24)

Student-t	Part 1	Part 2
Efficient	307.13 (10.83)	246.20 (9.58)
Non-Efficient	301.76 (8.95)	219.31 (4.53)

Exponential	Part 1	Part 2
Efficient	245.80 (15.10)	190.73 (8.30)
Non-Efficient	237.31 (6.61)	194.14 (4.13)

Table A.7. Total seconds spent on the estimation task, average across participants in each group (with SE).

Gaussian	Part 1	Part 2
Efficient	121.67 (5.58)	99.81 (4.49)
Non-Efficient	133.96 (8.88)	105.65 (6.26)

Student-t	Part 1	Part 2
Efficient	130.60 (8.06)	112.20 (6.95)
Non-Efficient	108.66 (4.45)	97.24 (4.00)

Exponential	Part 1	Part 2
Efficient	117.73 (10.12)	88.00 (5.01)
Non-efficient	111.66 (7.32)	90.93 (3.32)

Table A.8. Total seconds spent on the sampling task in the final 5 episodes, average across participants in each group (with SE).

Mean μ for odd ID number			Game 1	Game 2	Game 3
Part 1	LHS	risk free	3 ± 1	11	19 ± 1
	RHS	risky	3	11 ± 1	19
Part 2	LHS	risky	24	16 ± 1	8
	RHS	risk free	24 ± 1	16	8 ± 1

Table A.9. True mean values used in the data generating process. LHS: left hand side gaming machine; RHS: right hand side gaming machine. The risk-free gaming machine mean was the corresponding risky gaming machine mean ± 1 with 50% probability. For even ID number participants, the LHS and RHS gaming machines were the exact opposite.

	Color	IDs	Color	IDs	Color	IDs
Gaussian	blue	1, 7...	blue	2, 8...	cyan	3, 9...
Student-t	cyan		purple		blue	
Exponential	purple		cyan		purple	
Gaussian	cyan	4, 10...	purple	5, 11...	purple	6, 12...
Student-t	purple		cyan		blue	
Exponential	blue		blue		cyan	

Table A.10. Distributions assigned and background colors presented to participants. RGBs: blue = $(42, 82, 164)$, cyan = $(42, 125, 164)$, purple = $(102, 77, 164)$.

A.2 Chapter 6

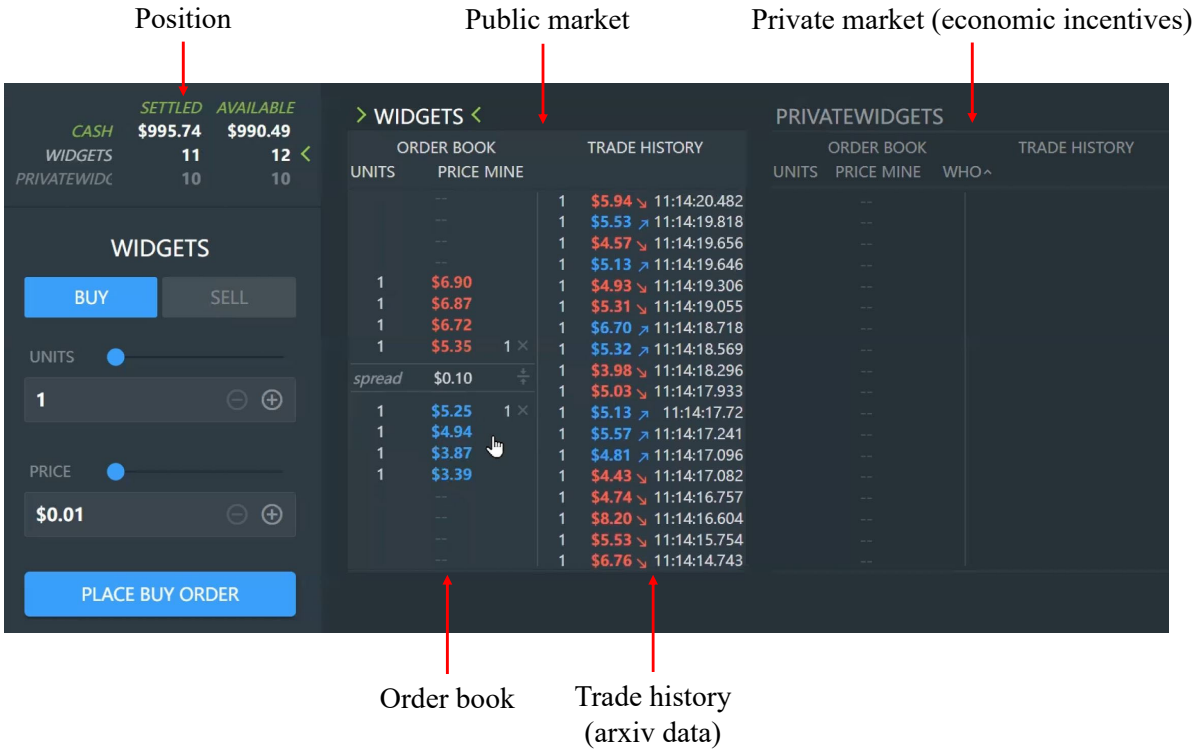


Figure A.2. Trading Graphic User Interface (GUI). A screenshot of the trading graphic user interface from Flex-E-Markets from the market maker's perspective. This is merely for an illustration purpose. All algorithmic trading activities are conducted at the backend using the Python APIs.

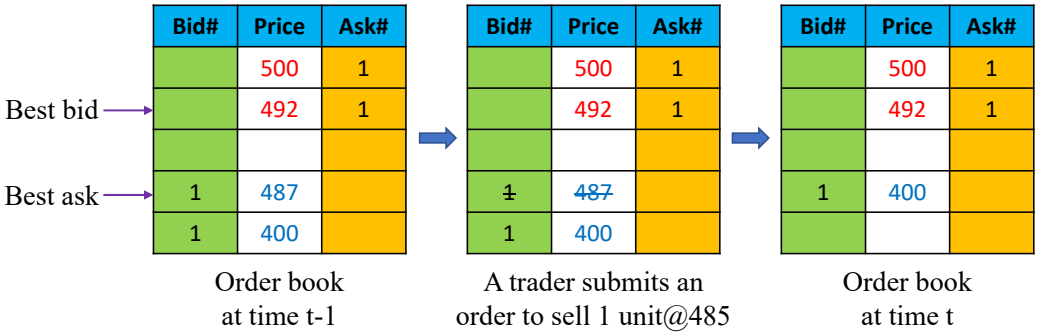


Figure A.3. Public market limited order book state transition. In the middle column, numbers in red are the ask prices, while numbers in blue are the bid prices. "Bid#" column shows the number of units for the bids while "Ask#" column shows the number of units for the asks. In this example, at time $t-1$, the best bid is 492 while the best ask is 487; hence the spread at $t-1$ is $492-487=5$. At t, a trader submits a sell order (ask) of one widget at 485. Because 485 is lower than 487, a trade will occur; the trader will sell one widget and receive 487. After the trade, the market will be left with three outstanding orders. The best bid will become 400 while the best ask remains at 492.

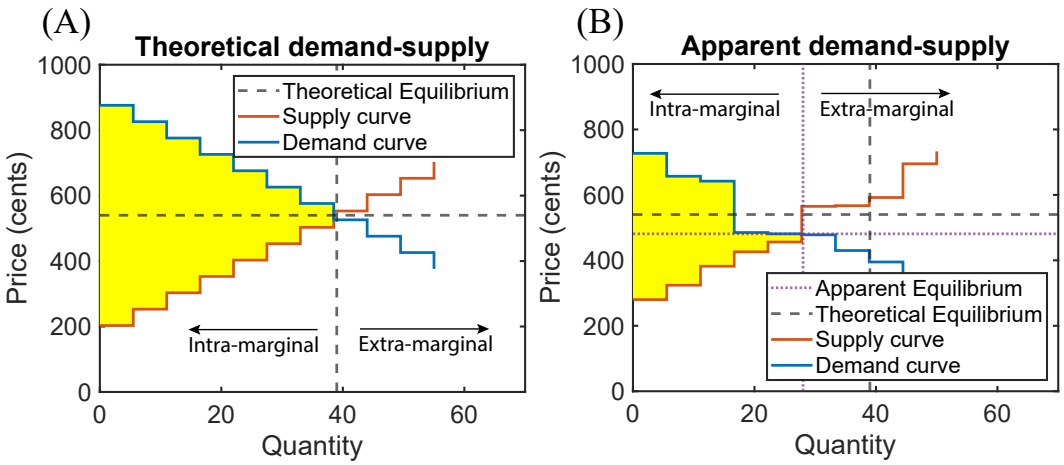


Figure A.4. Theoretical vs. apparent demand-supply. **(A)** The plot depicts the theoretical demand-supply curve (private signals). **(B)** we plot the apparent demand-supply curve based on the FIRST public order price submitted by each ZIT. The "true" equilibrium (called apparent equilibrium in Cliff (1997)) based on public order prices is different from the theoretical equilibrium when ZITs' profit margins are not zero. What's worse in our case is that agents submit a different price after each private order is consumed, making the demand-supply curve dynamic (hence the right-hand-side picture is merely for illustration).

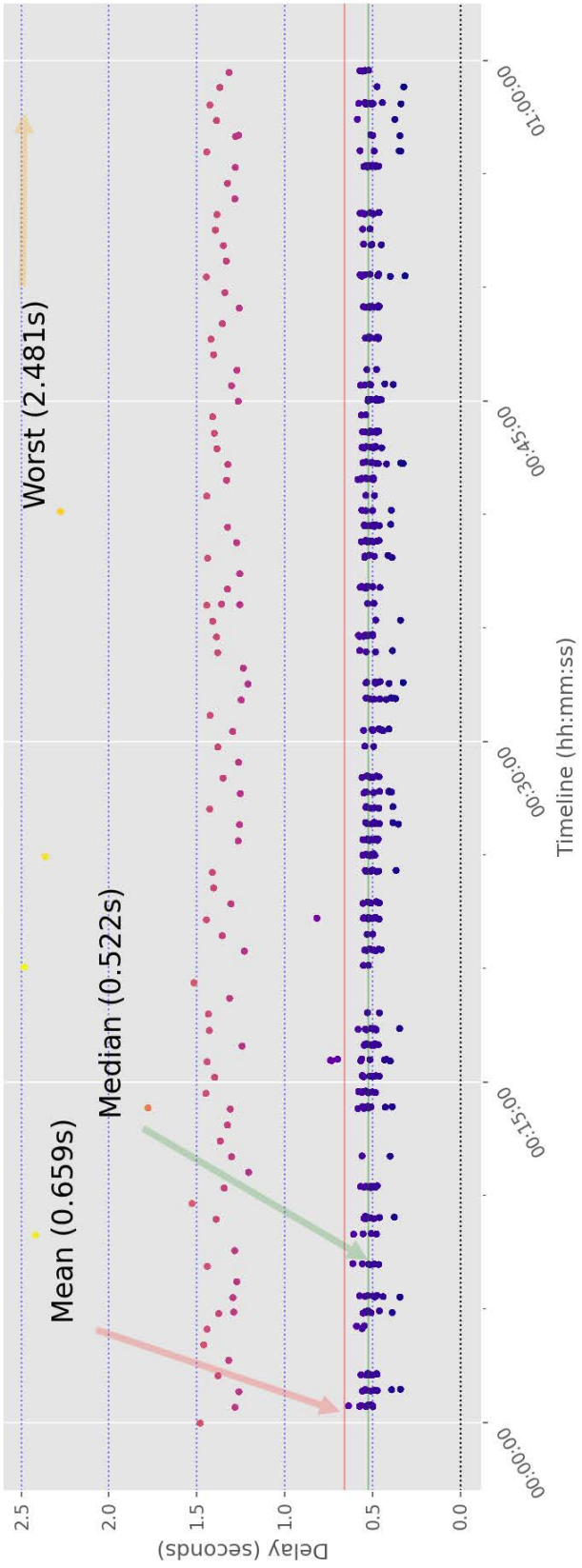


Figure A.5. Communication latency of one experiment

order 1 time	owner email	side	price	unit	order 2 time	owner email	side	price	unit	period
11:05:00.483000	u1@bmm	OrderSide.BUY	609	1	11:05:00.876000	u3@bmm	OrderSide.SELL	609	1	1
11:05:01.716000	u2@bmm	OrderSide.BUY	643	1	11:05:01.813000	u6@bmm	OrderSide.SELL	643	1	1

Table A.11. Order data example

A.3 SOFTWARE AND PACKAGES

I am grateful to the open-source community, institutions and companies that provide excellent software and packages to me for free.

- LaTeX(https://www.latex-project.org/)
- classicthesis by André Miede (http://www.miede.de)
- Python (https://www.python.org/)
- Numpy (https://numpy.org/)
- Pandas (https://pandas.pydata.org/)
- Scipy (https://scipy.org/)
- Matplotlib (https://matplotlib.org/)
- Seaborn (https://seaborn.pydata.org/)
- OpenAI-Gym (https://gym.openai.com/)
- WebSockets (https://websockets.readthedocs.io/)
- Unity (https://unity.com/products/unity-student)
- Unity WebSockets (https://github.com/jirihybek/unity-websocket-webgl)
- Newtonsoft.Json (https://github.com/jilleJr/Newtonsoft.Json-for-Unity)
- Matlab (https://www.mathworks.com, sponsored by University of Melbourne)
- makeLatexTable (https://au.mathworks.com/matlabcentral/fileexchange/77774-makelatextable)