Modeling Trading Strategies in Financial Markets with Data, Simulation, and Deep Reinforcement Learning

by

Megan Shearer

A dissertation submitted in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
(Computer Science and Engineering)
in The University of Michigan
2022

Doctoral Committee:

Professor Michael P. Wellman, Chair
Assistant Professor David Fouhey
Professor Uday Rajan
Professor Gabriel Rauterberg

You could call this selfhood many things.
Transformation. Metamorphosis. Falsity. Betrayal.
I call it an education.

—Tara Westover, *Educated*

Megan Shearer
shearerj@umich.edu
ORCID iD: 0000-0002-5680-3828

To my parents and little brother

ACKNOWLEDGEMENTS

First and foremost, I would like to thank my dissertation committee chaired by Michael Wellman and including David Fouhey, Uday Rajan, and Gabriel Rauterberg. My dissertation greatly benefited from their invaluable support and feedback. I would like to thank my advisor, Michael Wellman, for the guidance he offered me throughout my Ph.D. I am grateful that he allowed me to face the high and lows of graduate school in my own way, and that he let me pursue causes important to me in parallel with research. Under Mike's advising, my degree unfolded in a way that shaped me as a person and researcher. I would also like to thank Gabriel Rauterberg for his collaboration and mentorship. This dissertation would look very different without his collaboration. Gabe provided invaluable advice numerous times and I am extremely grateful for his unwavering positivity and encouragement. I am also grateful for Michael Barr and Christie Baer for their collaboration throughout my Ph.D.

I have also many collaborators and mentors outside of the University of Michigan. I am so grateful to Elaine Wah who recruited me to Michigan and gave me my first internship in grad school. She went above and beyond to support and mentor me, and my Ph.D. and confidence would look very different without her guidance. Elaine also introduced me to Abigail Jacobs who was another invaluable mentor the latter half of my degree. I will be forever grateful for her encouragement, for making me priortize my mental health, and for introducing to the show for geniuses to cope with grad school. Knowing I can always count on Elaine and Abbie for advice and encouragement has been one of the most comforting and stabilizing things throughout this process. I am also thankful for Tucker Balch and C.K. Chow for their collaboration and mentorship. I also found a friend and mentor in Raluca Havarneanu, who provided me with unconditional support and made my final internship a much more enjoyable experience.

During my time as a member of the Strategic Reasoning Group, I had many wonderful labmates and collaborators. I am thankful to have worked along side Mason

Wright, Frank Cheng, Xintong Wang, Max Smith, Zun Li, Yongzhao Wang, Christine Konicki, Katherine Mayo, and Madelyn Gatchel. Frank and Xintong were great friends as well and always supported me. I am also thankful for my collaboration with Max over the last two years. I also worked along side amazing research scientists and post docs: Mithun Chakraborty, Arunesh Sinha, and Thanh Nguyen.

I cannot express how grateful I am for my friendships with Harman Kaur, Preeti Ramaraj, and Lizzie Goeddel. Preeti and Harman were my life line and Lizzie kept me sane throughout this process. I found one of my most empathetic, fiery, optimistic, and loyal friends through Preeti. I was inspired by her dedication and passion while co-chairing ECSEL+ with her and will miss our weekly bike rides. Traveling, food, and drinks with Harman made hard parts of grad school fun. She is my partner-in-crime and one of my most adventurous, supportive, and loyal friends. My friendship with Preeti and Harman is best summed up by this quote:

> "There are some things you can't share without ending up liking each other, and knocking out a twelve-foot mountain troll is one of them."

I am also extremely grateful for my backpacking trips with Lizzie and her constant support and friendship. I can confidently say that no one else would want to do those trips with me, and each one let me disconnect and recharge. I am also immensely thankful for my friendships with Maddy Endres and Jaylin Herskovitz, both are amazing friends who I relied on for fun and support. I am also grateful for my other friends in Ann Arbor: Rob Goeddel, Allison McDonald, Kevin Loughlin, Fahad Kamran, and Liz Hou. Lastly, I am extremely grateful for my friends from Arizona, Maddie Sampsel, Molly Tatosian, and Audrey Thompson, who put up with me throughout my Ph.D. and who I cannot imagine life without.

Finally, I want to thank my family for loving and supporting me through this process. My mom, Jeanne Shearer, is my biggest supporter and best friend. I do not know where I would be without her willingness to take my calls and talk to me at anytime, and being a grounding force throughout my life. My dad, Michael Shearer, inspired me to get a Ph.D. and has always pushed me to think outside the box. The discipline and work ethic he instilled in me helped carry me through this degree. I am indebted to my parents for their unwavering support and encouragement to always step outside my comfort zone. I am also so grateful for my little brother, Trevor Shearer, who is always there for me when I need him. I do not know where I would be today without these three people.

TABLE OF CONTENTS

LIST OF FIGURES

Figure

LIST OF TABLES

Table

LIST OF APPENDICES

Appendix

ABSTRACT

Rapidly advancing algorithmic trading techniques and lagging financial market regulations have led to opportunities for traders to use these advancements to their own advantage. This dissertation explores trading practices that leverage superior market information or resources to benefit during certain market events. Studying trading activity during and around market events can provide insight into what may cause or exacerbate these events. I examine some common trading strategies that exploit an advantage to respond to market volatility before others are even aware of the event. I also analyze adversarial trading strategies that try to maximize combined profits between the market and an external contract whose value is dependent on a transaction-based financial market. These trading practices may increase market volatility, impact the profits of other market participants, or shift financial benchmarks.

This work analyzes the impact of various trading strategies in four studies. First, I use historical data to examine trading activity around mini flash crashes. I develop a two-step approach to detect a set of potential mini flash crashes, where each event is split into dynamic phases. I observe that some trading practices may be exacerbating these events and have negative effects on retail investors. Second, I use an agent-based simulation and empirical game-theoretic analysis to study ETF arbitrage. I also observe that when one traded asset experiences a mini flash crash, arbitrage increases market-wide volatility. However, the asset that initially experienced a mini flash crash recovers faster when there is arbitrage. Third, I study benchmark manipulation in an agent-based simulation. I expand the market model to include a transaction-based benchmark, and examine three ways of constructing benchmark manipulation strategies. One form of strategy is hand-crafted and heuristic and the others are derived by deep reinforcement learning. I observe that all three manipulators successfully move a benchmark by submitting aggressively priced orders during the trading period. The other trading agents benefit from benchmark manipulation because the manipulator takes a loss in the market to gain a larger profit through the benchmark. Benchmark manipulation negatively affects the parties on the other side of the benchmark contracts. Lastly, I combine data-driven and computation

approaches in a pilot study of close price manipulation. I use a simulated market with a close auction and continuous market. I use historical market data to generate a conditional, synthetic close auction order stream. This order stream is injected into the close auction. A manipulator that learns its trading strategy through deep reinforcement learning tries to manipulate the close price from the continuous market. In this initial implementation, the manipulator is unable to influence the close price.

CHAPTER I

Introduction

Two events of around ten years ago led to public attention and scrutiny and inspired various research studies. The first event was the Flash Crash of May 6, 2010, when prices of many U.S. stocks changed by as much as 60% in 5 minutes, and approximately 2 billion shares traded between 14:40 and 15:00, but most securities reverted back to near their original prices within 15 minutes (U.S. Securities and Exchange Commission, 2010). The second was the London Interbank Offered Rate (LIBOR) scandal of 2011, where contributors manipulated this benchmark by self-reporting inaccurate data to the party that calculated the benchmark (Duffie and Dworczak, 2018; Gellasch and Nagy, 2019). LIBOR is an estimate of the rate at which banks can borrow from each other and supports more than $300 trillion worth of loans around the world. Several major banks were implicated in schemes to manipulate LIBOR in the last decade, and criminal charges have been brought against over twenty individuals in the U.S. and U.K. since 2015 (McBride, 2016). Both the Flash Crash and LIBOR Scandal made market regulators rethink trading practices and benchmarks.

While the Flash Crash and LIBOR Scandal occurred over ten years ago, continuing market patterns cause even more scrutiny around what initially caused and exacerbated these events and which traders were adversely impacted. High market volatility persists today (Duguid and Platt, 2022; Kaplan, 2022; Platt and Rennison, 2022) Recent market volatility has led to fears that another larger volatility event, like the Flash Crash, is imminent and could contribute to the next recession (Heath, 2018; Kim, 2018). There is also a growing concern for the vulnerability of certain market participants during larger volatility events because of the recent increase in retail volume and market volatility (Horstmeyer, 2020; Wursthorn et al., 2020). The belief that another larger volatility event could happen stems from the frequency of smaller volatility

events (Bowley, 2010); some even speculate that these smaller events occur over twelve times per day (Farrell, 2013). These so-called *mini flash crashes* possess the same characteristics as the Flash Crash, but in a single symbol with a smaller magnitude and shorter duration. Given that mini flash crashes are frequent, miniature renditions of the Flash Crash, similar patterns in order and trade activity likely exist between these two types of events. Mini flash crashes are typically defined as occurring in one symbol, but certain trading practices may amplify volatility across multiple symbols (Ben-David et al., 2015; Pagano et al., 2019). Studying behavior during mini flash crashes can help determine what intensifies a larger volatility event like the Flash Crash.

The second pattern that continues to happen is the manipulation of other financial benchmarks across various markets. February 2018 saw accusations of manipulation in the Chicago Board Options Exchange (CBOE) *Volatility Index* (VIX), a measure of U.S. stock market volatility based on the cost of buying certain options (Banerji, 2018). In the wake of the LIBOR scandal, regulators, academics, and market participants lobbied for a transaction-based replacement for LIBOR, such as the Secured Overnight Finance Rate (SOFR) or the U.S. Dollar Intercontinental Exchange (ICE) Bank Yield Index (Duffie and Dworczak, 2018; ICE Benchmark Administration Limited, 2019). Whereas it may be harder to manipulate transaction-based benchmarks, it is still possible, as in the alleged manipulation of the VIX in 2018 and the World Markets/Reuters Closing Spot Rates (WM/R FX rates) in 2014 (Boyle, 2014). There are also smaller benchmarks like closing prices which may be subject to manipulation as well because they inform asset fund valuations, company swaps, and quarterly reports. A closing price may be more susceptible to manipulation than a market-wide benchmark, like the VIX, because there is less trading volume in a single symbol.

Examining trading practices around major market events can provide insight into what may cause or exacerbate these events. In Section 1.1, I discuss common and emerging trading practices which exploit market information and microstructure. In Section 1.2, I outline my methodology to study the effects on the market of these trading strategies. In Section 1.3, I provide an overview of work in this dissertation.

1.1 Trading Strategies that Leverage Market Information and Microstructure

I explore numerous trading strategies that take advantage of certain market information and microstructure. Each of these strategies capitalizes on market information and microstructure to increase profits, and some of them could qualify as adversarial or manipulative. Intentionally adversarial or not, each strategy has the potential to deteriorate the quality of the market. These trading practices may increase market volatility, decrease the profits of other market participants, or shift financial benchmarks.

Intermarket Sweep Orders *intermarket sweep orders* (ISO) are limit orders that execute on a trading venue even if other venues are listing better prices, so this order type changes the responsibility for best execution from the trading venue to the market participant (U.S. Securities and Exchange Commission, 2005). These orders are commonly used to exit a large position or take out multiple price levels in the order book. Participants can also use these orders for a quicker order execution (Chakravarty et al., 2012). During a mini flash crash, it makes sense for market participants to submit ISOs to exit their position when the price is rapidly dropping. However, it is interesting to examine ISOs immediately prior to mini flash crashes. Since ISOs allow a participant to immediately take out levels of a trading venue's order book, even when other venues list better prices, these orders may trigger or amplify the price drop. ISOs may cause the price to drop enough to activate retail investors' *stop-limit orders*, orders set to automatically trade when a symbol reaches a specified price. Another participant might hope to trade with retail investors, because they are typically slower to act and less informed than other market participants. Submitting ISOs immediately preceding a mini flash crash may signal that some traders may attempt to incite the event to trade with stop-limit orders.

Internalization of Retail Volume In the U.S. equities market, *payment for order flow* is the practice in which brokers purchase investors' orders, then execute these orders against other brokers' orders (Battalio and Holden, 2001; Parlour and Rajan, 2003). *Wholesalers* are brokers who purchase retail investors' orders through payment for order flow, then internalize a portion of these orders. Internalization is when a broker executes an order against its own account. Wholesalers often offer subpenny

price improvement on internalized retail orders to attract retail order flow (Levine, 2018). One might question why wholesalers pay for retail volume, then offer a better price to the retail investors than if their order traded on an exchange.

Wholesalers are better informed than most retail investors, which may be why wholesalers internalize retail volume with price improvement. It is possible that they could notice a change in price before the retail investors and attempt to profit off of it. Stop-limit orders are usually not marketable when retail investors place them, so a lot of these orders are passed to wholesales through payment to order flow. During an event, like a mini flash crash, that triggers a large number of stop-limit orders, wholesalers have the opportunity to internalize these orders. If a wholesaler chooses to internalize a stop-limit order, then they could buy this order for lower price at the bottom of a mini flash crash. A change in price direction can signify to a wholesaler that the price may revert back to its original level, so it would be profitable to buy any available sell orders at a lower price. Wholesalers are not able to definitively know if the price of the asset will recover, so they are taking a short-term risk by potentially buying retail volume at the bottom of a mini flash crash. If sell stop limit orders are internalized at the lowest price of a mini flash crash, then these events may adversely affect retail investors.

ETF Arbitrage Exchange-traded funds (ETF) are baskets of assets that trade on the stock market. ETFs are commonly associated with a *market index*, which represent a portfolio of securities designed to track some aspect of the stock market. *Index-based ETFs'* underlying securities are stocks that compose a market index. Some examples of market indexes are the Standard and Poor's (S&P) 500, Dow Jones Industrial Average (DJIA), and the Volatility Index (VIX). The S&P 500 and the DJIA track market performance, while the VIX tracks market volatility. An *index-based ETF* includes underlying securities from a corresponding market index.

While an ETF is designed to track the index, how it trades as a symbol on the stock market determines the actual price of the ETF. When the market is open, participants can simultaneously observe the trading price of the ETF and calculate the index. Any disparity composes a potential arbitrage opportunity between the ETF and the underlying securities. Arbitrage trading can help an ETF's price track its corresponding market index (Antoniewicz and Heinrichs, 2014). It may also introduce or reinforce other dependencies. For example, arbitrage may tether the price volatility of the ETF's underlying symbols (Ben-David et al., 2015). The rise in market-wide

price volatility in recent years has raised concerns about economic growth and investor trust (Carlson, 2019; Li, 2019). Price co-movements in ETF portfolios have led some to question the role ETFs play in amplifying volatility (Ben-David et al., 2015; Pagano et al., 2019). Using an ETF arbitrage strategy may transmit mini flash crashes from one underlying symbol to others in the portfolio. ETF arbitrage may also increase the profits of the arbitrageurs, but decrease the profits of other traders.

Strategy Generation with Deep Reinforcement Learning With the recent advances in deep reinforcement learning (DRL) and its natural application to trading because of its goal to maximize an agent's reward, using DRL to generate trading strategies is a topic of immense interest in both academia and industry. Using DRL is not a trading strategy within itself, but rather a tool to generate trading strategies which capitalize on market information to maximize the agent's reward. While utilizing strategies generated by DRL provides an opportunity to increase the profits of the trader deploying those strategies, it also may have adversarial impacts on the overall market.

A participant may receive a profit from market activity other than their own performance. For example, if the participant is party to a contract whose value is dependent on an asset's market activity, then the participant would receive a profit from the contract and their own trading activity in the market. The participant could design their reward function to include profits gained from this external contract. The participant may then submit orders they otherwise would not in the DRL algorithm's attempt to increase its reward. This could arguably fall under the definition of market manipulation because the participant submits orders it otherwise wouldn't to increase their own profits. A reason this is a particularly interesting question now, as using DRL to generate trading strategies increases in popularity, is currently the definition of market manipulation is that the market participant has the *intent* to manipulate (U.S. Securities and Exchange Commission, 1934). One may argue that with DRL strategy generation, a participant did not intend to manipulate the market, but the DRL algorithm chose the strategy. I explore how choices in the reward function may influence the trading strategies generated by a DRL algorithm. If this is the case, the definition of market manipulation should expand to include a market participant's algorithm design.

Benchmark Manipulation A market participant may be party to an external contract based on a *financial benchmark*, which is a summary statistic over market variables. If this financial benchmark is based on transactions in a market, then the participant's profit depends on both the valuation of the benchmark and their market activity. Benchmarks are employed by market participants for various purposes, including reference measures for asset values (e.g., the S&P 500), interest rates (LIBOR), and market volatility (VIX); to define derivative instruments; or as price terms in contracts (Gellasch and Nagy, 2019). Some participants may have large stakes in benchmark values, so they have incentives to attempt to manipulate benchmarks. LIBOR has been particularly vulnerable to manipulation because it is calculated using self-reported data provided by parties with conflicts of interest regarding the benchmark's value (Duffie and Dworczak, 2018; Gellasch and Nagy, 2019). In the wake of the LIBOR scandal, regulators, academics, and market participants lobbied for a transaction-based replacement for LIBOR, such as the Secured Overnight Finance Rate (SOFR) or the U.S. Dollar Intercontinental Exchange (ICE) Bank Yield Index (Duffie and Dworczak, 2018; ICE Benchmark Administration Limited, 2019). Whereas it may be harder to manipulate transaction-based benchmarks, it is still possible, as in the alleged manipulation of the VIX in 2018 and the World Markets/Reuters Closing Spot Rates (WM/R FX rates) in 2014 (Boyle, 2014).

A market participant could attempt to shift a benchmark by submitting orders it otherwise would not. A participant could submit buy orders at prices higher than it thinks the asset is valued in an attempt to increase the benchmark. In this case, the trader takes a loss in the market to increase its external profits through the benchmark. A participant could also use deep reinforcement learning strategy generation to manipulate the benchmark. It could then include its contract holdings from the benchmark in their reward function and metrics influencing the benchmark in their state space. The participant may then submit orders they otherwise would not in the DRL algorithm's attempt to increase its reward by influencing the benchmark. The design of a transaction-based benchmark should aim to be robust to these types of manipulation.

Close Price Manipulation The *close price*, the last trade price in a market trading period, is an important financial benchmark for contracts, derivatives, and reference prices (Alexander and Cumming, 2020). Some believe close price manipulation frequently happens, but is rarely detected in the US market (Comerton-Fordea and

Putninš, 2011). The close is potentially very lucrative benchmark to manipulate because many contracts utilize close prices. A participant could access a lot of data to learn their trading policy, given that every asset in a market has a close price each trading day. It also may be easier to manipulate the close price than other transaction based benchmarks because a manipulator would only need to trade one symbol, rather than many in a benchmark like a market index.

The close price is determined at the end of a trading period, and is set to price which will maximize the number of shares paired in the US stock market and close auction. During the trading period, traders do not have access to order book information for the close auction, so the exchange hosting the close auction publishes imbalance messages. *Imbalance messages* are periodically released statistics which provide information about the close auction and predict where the price will close. A participant could consider imbalance message information while using DRL to generate trading strategies to manipulate the close price of an asset.

1.2 Data-Driven & Computational Approaches

Which approach to take when studying trading strategies in financial markets is a complex question. Many prior works in Finance rely on historical data to study behavior in financial markets. Historical data supplies opportunities to analyze real market events. However, historical data can be difficult to access and new trading strategies are difficult to test with this type of data. Another approach to analyzing trading strategies is computational modeling. Using agent-based models and simulation allows an analysis of the interactions and decisions of different trading agents, though this type of computational approach can oversimplify some aspects of the market and can be difficult to extend to a real financial market. Combining these two approaches could combine the strengths of each method to better analyze the success and impact of different algorithmic trading strategies.

Data-Driven Approach Many use historical data as a tool to study behavior in financial markets. Historical data allows an analysis of actions taken in the real market. However, historical data is hard to access. The easiest forms of data to access for academics are trade and quote data. Most data sets do not label which participants submit which orders or take part in a transaction. This makes it very difficult to study trading strategies with historical data. Many studies that use historical data

rely on proxy methods to identify participant types (Boehmer et al., 2017; Shearer, 2020), while other studies use datasets tagged by aggregated participant type (Bellia et al., 2020; Brogaard et al., 2018). Another limitation of historical data is that it is invariable and final. If a new participant is injected into a stream of historical data, then once the new participant submits an order or trades, then all of the market activity which originally followed is arguably irrelevant. When a new order or trade enters the market, other participants will react to it and potentially change their future behavior. Thus, it is infeasible to study new trading strategies with solely historical data.

Computational Approach Agent-based modeling (ABM) and simulations are computational approaches that allow an analysis of the dynamics and reactions of trading agents in a market setting. An ABM is populated with trading agents who trade based on their own beliefs and strategies. These agents are highly adaptive to a changing environment and respond to additional market information. Each agent aims to maximize its own profit in these models. Thus, this approach is more naturally suited to study the success and effects of different trading strategies. However, ABM can oversimplify market environments. ABMs can pare down the market environment to a point where it is difficult to extend those results back to a real market setting. Thus, while ABM is a useful tool for studying agent interactions in financial markets, it may be difficult to address some questions in real market settings with this approach.

Combining Data-Driven & Computational Approaches I propose to combine data-driven and computational approaches to reap the rewards of both methods. Prior work uses generative adversarial networks (GANs) in an attempt to learn realistic order streams (Li et al., 2020). GANs can utilize the structure of the market to capture patterns in order history. I generate synthetic order streams by using historical order data to train a GAN model similar to that of Li et al. (2020). A synthetic order stream aims to learn realistic order patterns to replicate in a simulation. A synthetic order stream can incorporate incoming market information to generate orders. This addresses one limitation of historical data because this data will not become stale with additional activity entering the market. I inject this data into an agent-based market model to determine if an agent that uses deep reinforcement learning can manipulate the market. Using this synthetic data in a simulation tackles the limitation of agent-based modeling by supplying a more realistic market model. Thus, the

findings may apply more easily to real financial markets.

1.3 Dissertation Overview

My thesis concentrates on the effects of trading strategies that capitalize on market features during volatility events or to manipulate a financial benchmark. I utilize data-driven and computational approaches individually and in tandem throughout my thesis to study the success of certain trading strategies. I also analyze the impact of these strategies on market volatility and welfare.

Chapter II presents a study using historical data to examine trading activity around mini flash crashes. I develop a two-step approach to detect a set of potential mini flash crashes, where each event is split into dynamic phases. I then analyze the set of mini flash crashes to determine what trade activity causes or exacerbates these events. Starting a minute before the price drop begins, I find the proportion of ISOs gradually increases. I also observe that volume of potential retail sell volume spikes by 278% around the lowest price of the mini flash crash. I began this work when I was a Visiting Research Fellow at the Investor's Exchange (IEX). This work has also been reported in another paper (Shearer, 2020).

The rest of my thesis employs agent based-modeling to model and study agent interactions in financial markets. I introduce the agent-based financial market models I use in Chapter III. I use two market simulation platforms, Market-Sim and ABIDES. These two models are very similar in their trading mechanisms and asset valuations. I also discuss various background strategies I employ in both market models. I present a new framework which allows agents to generate trading strategies through DRL in Market-Sim. Lastly, I discuss how I use empirical game-theoretic analysis to compute equilibria in simulated market environments.

In Chapter IV, I present work that uses agent-based simulation and empirical game-theoretic analysis to study ETF arbitrage. I model a simplified index-based ETF with a portfolio that contains two underlying symbols. I find when arbitrageurs actively trade, background traders' surplus increases when they select more conservative strategies and decreases if they select more competitive strategies. I also note that when one symbol experiences a mini flash crash, the other symbol experiences a price change in the opposite direction when arbitrageurs actively trade. Last, the size of the mini flash crash is more dependent on the competitiveness of the background traders than the arbitrageurs, but the recovery of the mini flash crash is faster when arbitrageurs are

present. This work is in collaboration with researchers at JP Morgan AI Research and Georgia Tech, and was also presented in the ICAIF conference proceedings (Shearer et al., 2021).

In Chapter V, I present a study on benchmark manipulation in an agent-based simulation. I expand the market model to include a transaction-based benchmark. I examine three types of benchmark manipulation, one heuristic strategy and two strategies learned through DRL. I find that the manipulators successfully move a benchmark by submitting aggressively priced orders during the trading period. The DRL manipulator outperforms the heuristic manipulator. The other trading agents benefit from benchmark manipulation because the manipulator takes a loss in the market to gain a larger profit through the benchmark. This manipulative strategy negatively affects the parties on the other side of the benchmark transaction.

Chapter VI discusses combining data-driven and computation approaches to study close price manipulation. I expand a market simulator to include a close auction and imbalance messages. I use historical market data in an attempt to generate a realistic, synthetic close auction order stream in an agent-based simulation. I then implement a DRL manipulator in the continuous market to determine if it can shift the close price for its own gain. This work was done while I worked at the Financial Industry Regulatory Authority (FINRA).

In Chapter VII, I summarize the contributions of my dissertation.

CHAPTER II

The Phases and Catalysts of Mini Flash Crashes

Mini flash crashes are rapid volatility events reminiscent of the May 6, 2010 Flash Crash in the US stock market. The frequency of mini flash crashes and their similarity to the Flash Crash provide an avenue to study patterns in order and trade activity during smaller volatility events to gain greater insight into potential activity during larger volatility events. I find a set of potential mini flash crashes using a two-step process. I then divide each mini flash crash into phases, where I define a phase as a stage with the same price directionality. Splitting each mini flash crash into phases facilitates a more dynamic and granular examination of order and trade activity during different stages of a mini flash crash. Using Daily TAQ data, I find that the proportion of sell Intermarket Sweep Orders increases before the price begins to drop. This result suggests that some participants may exacerbate mini flash crashes with certain order and trade practices. Using off-exchange subpenny trades as a proxy for retail activity, I find a significant increase in volume potentially initiated by retail sell orders executed at or near the lowest price of the mini flash crash, which may indicate that retail investors could be particularly subject to harm during these events.[1]

2.1 Introduction

During the Flash Crash on May 6, 2010, prices of many U.S. stocks changed by as much as 60% in 5 minutes and approximately 2 billion shares were traded between 14:40 and 15:00, but most securities reverted back to near their original prices within 15 minutes (U.S. Securities and Exchange Commission, 2010). Some postulate that a

[1]Wharton Research Data Services (WRDS) was used in preparing this research. This service and the data available thereon constitute valuable intellectual property and trade secrets of WRDS and/or its third-party suppliers. I gratefully acknowledge Investors Exchange for research support.

larger volatility event similar to the Flash Crash could happen again, but that prices might not swiftly revert, which could contribute to the next financial recession (Heath, 2018; Kim, 2018). There is also a growing concern for the vulnerability of certain market participants during larger volatility events because of the recent increase in retail volume and market volatility (Horstmeyer, 2020; Wursthorn et al., 2020). The belief that another larger volatility event could happen stems from the frequency of smaller volatility events (Bowley, 2010); some even posit that these smaller events occur more than twelve times per day (Farrell, 2013). These so-called *mini flash crashes* possess the same characteristics as the Flash Crash, but in a single symbol with a smaller magnitude and shorter duration. While mini flash crashes occur in a single symbol, multiple symbols can experience mini flash crashes simultaneously. Given that mini flash crashes are frequent, miniature renditions of the Flash Crash, similar patterns in order and trade activity likely exist between these two types of events. Studying behavior during mini flash crashes can help determine what intensifies a larger volatility event like the Flash Crash.

Mini flash crashes do not have a well-established definition, so I develop a phase-based approach to analyze these events. I define a *phase* as a stage with the same price directionality. Many prior studies assume the duration of the price drop stage happens over a fixed time interval (Aquilina et al., 2018; Braun et al., 2018; Golub et al., 2017; Johnson et al., 2013). However, I find that the severity and duration of each event is unique. Therefore, I divide each mini flash crash into phases, which facilitates a more dynamic and granular comparison of order and trade activity across these events. I specify three phase transition points: the start, lowest price, and end of the mini flash crash. The start happens when the price first begins to rapidly drop, whereas the end is reached when a symbol's price stabilizes after reverting. More formally, the phases are:

Phase I: Ten seconds preceding the start of the crash,

Phase II: The start of the crash to the lowest price,

Phase III: The lowest price to the end of the reversion,

Phase IV: Ten seconds following the end of the reversion.

To determine if activity differs in events where the price drops but does not revert, I also collect a set of events with minimal reversion after the lowest price. I define an

abrupt price disruption to be this type of event, which is similar to a mini flash crash in every aspect, but with less price reversion. Comparing activity during mini flash crashes to abrupt price disruptions provides insight into behavioral differences when the price reverts.

I analyze trade and order activity during each phase using Daily TAQ data. I study Intermarket Sweep Orders (ISO), which are limit orders that can execute on a trading venue even if other venues are quoting better prices. It is the broker's responsibility to simultaneously send ISOs to other venues with prices better than the best price quoted on the given venue. A venue typically holds the responsibility to verify that an order transacts with the best displayed quote, but an ISO transfers this responsibility to the broker.[2] Similar to the conclusions of Braun et al. (2018) and Golub et al. (2017), I find that the proportion of sell ISO volume to total market volume gradually increases about 30 seconds before the price begins to drop. This result suggests that some market participants potentially exacerbate mini flash crashes.

Using TAQ data and a method similar to Boehmer et al. (2017), I also analyze off-exchange subpenny trades at prices common for retail volume as a proxy for retail activity, since retail volume is not directly identifiable using TAQ data. Other investors also trade off-exchange at subpenny prices, so this proxy method further isolates subpenny prices more frequent for trades initiated by retail orders. Given the short duration of mini flash crashes, most retail investors likely lack the ability to react fast enough to directly trade during these events. However, retail investors may use *stop-limit orders* which are orders set to automatically trade when a symbol reaches a specified price. Stop-limit orders can potentially react quickly enough to transact during mini flash crashes and abrupt price disruptions, so many of these off-exchange subpenny trades I identify in my dataset as potentially originating from retail investors are likely stop-limit orders. My results indicate that the volume of trades potentially initiated by retail sell orders immediately after the lowest price of a mini flash crash is 278% higher than the average volume of potential retail sells throughout the price drop. The volume of potential retail sells at or near the lowest price of an abrupt price disruption is a tenth of the volume of potential retail sells at or near the lowest price of a mini flash crash. This implies that during mini flash crashes, potential retail sell orders are more likely to be internalized at or near the lowest price. Therefore, retail investors may potentially be subject to harm during

[2]To adhere to Regulation National Market System Rule 611, orders must transact with quotes at the best displayed price across exchanges, rather than the best price at a single trading venue.

mini flash crashes by selling at lower prices.

This chapter is organized as follows. Related work is discussed in Section 2.2. Section 2.3 describes the order and trade data sources I use to study market participant activity. Section 2.4 discusses my methodology for defining and detecting mini flash crashes. Section 2.5 presents my analyses of the detected potential mini flash crashes. Section 2.6 presents results on order and trade patterns during each phase. I conclude in Section 2.7.

2.2 Related Work

Mini flash crashes lack a formal definition, so how a chapter defines mini flash crashes creates a natural divide within prior work. Nanex (2011) provides one of the most widely-used definitions—employed in studies by Braun et al. (2018), Golub et al. (2017), Goncalves et al. (2019), Johnson et al. (2013), and Laly and Petitjean (2020)—which defines a mini flash crash as an event where the price of a symbol must monotonically change after at least 10 trades by at least 0.8% within 1.5 seconds. Other studies by Bellia et al. (2020), Christensen et al. (2017), and Tee and Ting (2020) less formally define mini flash crashes, and instead build mathematical models to detect significant price deviations over short time horizons. Other prior work by Aquilina et al. (2018) and Tse et al. (2018) uses a third definition, which defines a mini flash crash as an event where the price change exceeds a pre-determined threshold, the price reverts back to within 50% of the price in a given time frame, and there is high trading volume during the price change. This chapter's definition of a mini flash crash closely follows this third definition, because price reversion and abnormally high trading volume were defining characteristics of the Flash Crash. Previous studies consider mini flash crashes, also referred to in the literature as Ultra Extreme Events, Extreme Price Movements, and Momentum Ignition events, to include both upward and downward movements in price, though these two events are typically analyzed in two separate datasets (Aquilina et al., 2018; Braun et al., 2018; Brogaard et al., 2018; Christensen et al., 2017; Golub et al., 2017; Johnson et al., 2013; Laly and Petitjean, 2020; Tse et al., 2018). This chapter only considers significant downward movements in price, because different activity patterns could trigger or intensify downward versus upward movements.

My phase-based approach to detect and study mini flash crashes distinguishes my work from others. Most other studies assume the drop of a mini flash crash happens

over a fixed time interval (Aquilina et al., 2018; Braun et al., 2018; Brogaard et al., 2018; Golub et al., 2017; Johnson et al., 2013). However, each mini flash crash is unique, and dividing a mini flash crash into phases can thereby yield a dynamic and granular comparison of order and trade activity across these events.

Other studies used a wide range of datasets to draw conclusions about what contributes to mini flash crashes. Prior work frequently considers high-frequency traders (HFT), who are defined as high-speed, proprietary traders. Aquilina et al. (2018) study orders and trades by market participant type in the UK secondary equity market, and find that non-HFT participants trade the most at the beginning of the price change in mini flash crashes, but HFTs potentially exacerbate the price change once it begins by trading in the direction of the change. Braun et al. (2018) use TAQ data timestamped to the second, and conclude that some mini flash crashes are caused by a large, aggressively-priced order that takes out multiple price levels. They also determine that fewer than 40% of mini flash crashes recover completely in price. Bellia et al. (2020) analyze trade and order data labeled by participate type on the Euronext stock market, and find that HFTs take liquidity during mini flash crashes. Brogaard et al. (2018) use trade data provided by NASDAQ marked as HFT or non-HFT, finding that HFTs provide liquidity to non-HFTs during mini flash crashes. Goncalves et al. (2019) studies executed orders on NASDAQ which use a speed bump, and find that this delayed order type reduces the number of mini flash crashes on NASDAQ and improves the liquidity provisions by HFTs. Golub et al. (2017) study publicly reported trade and quote data, and conclude that ISOs can cause mini flash crashes.

Another vein of prior work simulates mini flash crashes to study activity that might contribute to these events. Bayraktar and Munk (2017), Brinkman (2018), and Leal and Napoletano (2019) study *trend-following strategies*, which are short-term trading strategies in response to market volatility by trading in the same direction as the price movement to avoid adverse selection (Bhansali and Harris, 2018). These studies find that trend-following activity can contribute to volatility events. If proprietary and high-speed traders use trend-following strategies during mini flash crashes, then they could potentially drive down the price.

The majority of prior work on mini flash crashes focuses on exacerbations to the price drop, though a prior study by Madhavan (2012) posits that retail investors were possibly harmed in the reversion of the Flash Crash. I study both activity before the price drop and retail activity during the reversion of a mini flash crash, which can help provide insight as to when and how retail investors are potentially harmed during

these events.

2.3 Data

This chapter utilizes trades and National Best Bid and Offer (*NBBO*) quote updates from TAQ data timestamped to the microsecond. The *NBBO* consists of the National Best Bid (*NBB*) and the National Best Offer (*NBO*), which are the best available bids and offers displayed across all exchanges' limit order books, respectively. TAQ data provides the benefit of information from all venues, though has the disadvantage that market participant and order information are unknown. This dataset facilitates detection of potential mini flash crashes and analysis of trade patterns across all trading venues.

I search for mini flash crashes between June 2017 and June 2018, and consider quotes and trades between 9:30 and 16:00 using TAQ data timestamped to the microsecond. However, I consider mini flash crashes only between 9:45 and 15:55, because the first fifteen minutes and last five minutes of the day are significantly more volatile than the rest (Bain et al., 2014). To exclude illiquid symbols, I restrict my analysis to common stocks and ETFs, and exclude test symbols, penny stocks, symbols with fewer than 500 trades that day, and symbols with volume less than 100,000 shares that day.[3]

Similar to Wah et al. (2019), I filter quotes and trades to exclude any irregular trading activity. I only consider quotes where $NBO \in [\frac{1}{3}NBB, 3NBB]$. A *crossed market* occurs when $NBO < NBB$ and is extremely rare, though they can occur during high volatility (Favreau and Garvey, 2019). I exclude the symbol BRK.A (Berkshire Hathaway Class A stock) because it is irregular in both price and quantity traded. I remove potentially erroneous executions by omitting trades that are more than 10% outside of the *NBBO*, or when the price is not within the range $[NBB - 0.1M, NBO + 0.1M]$, where $M = \frac{1}{2}(NBB + NBO)$ is the midpoint. I exclude trades with correction indicators not equal to 0 or 1, trades with sale condition codes B, C, G, H, L, M, N, O, P, Q, R, T, U, V, W, Z, 4, 5, 6, 7, 8, or 9, and transactions on the Chicago Stock Exchange.[4]

[3] A penny stock is defined as a symbol whose average *NBBO* midpoint between 9:30 and 16:00 is less than $5 over the past five days (see https://www.sec.gov/fast-answers/answerspennyhtm.html).

[4] For more information on the sale condition codes, please refer to the TAQ Client Specification at http://www.nyxdata.com/data-products/daily-taq.

2.4 Detection Methodology

I define a *mini flash crash* as a sudden dislocation in price and high trading volume, followed by a swift price reversion. This definition is similar to the one used in Aquilina et al. (2018) and Tse et al. (2018). I calculate the *price* of a symbol as the midpoint of the National Best Bid and Offer (*NBBO*). The *NBBO* consists of the National Best Bid (*NBB*) and the National Best Offer (*NBO*), the best available bids and offers displayed across all exchanges' limit order books. However, the *NBBO* updates irregularly based on when trades and quotes affect its value, so I sample prices at fixed time intervals. Figure 2.1 depicts a mini flash crash in the symbol EOG around 12:22:20 on November 1, 2017, where the price quickly drops, but then reverts. After the Flash Crash, a *limit up-limit down* (LULD) threshold was introduced to prevent transactions outside of specified price bands (U.S. Securities and Exchange Commission, 018b). This mini flash crash in EOG hit a LULD threshold, and no transactions occurred for about 0.4 seconds between about 12:22:30.1 to 12:22:30.5. This 0.4 second trading break is better illustrated in Figure 2.3, and it is worth noting that the price dislocation might have been more severe without the LULD threshold. In EOG, the reversion took a little longer than the drop, but overall the event was quick and took about 1.8 seconds. The general shape of other mini flash crashes tends to be similar to that of EOG, but these events vary in the severity and duration of the drop and reversion. Therefore, after finding a potential mini flash crash I divide it into phases to facilitate a more in-depth analysis. The phase transition points are the start of the crash when the price begins to consistently decrease, the lowest price, and the end of the reversion when the price stops consistently increasing. I define the *phases* of a mini flash crash as:

Phase I: Ten seconds preceding the start of the crash,

Phase II: The start of the crash to the lowest price,

Phase III: The lowest price to the end of the reversion,

Phase IV: Ten seconds following the end of the reversion.

Algorithm 1 outlines my two-step process to detect potential mini flash crashes. The first step, *Detection*, performs a general sweep over 10-second intervals, searching for a large price drop, a local minimum price, a price reversion, and high trading volume. The second step, *Phase Identification*, uses prices sampled at 10-millisecond intervals

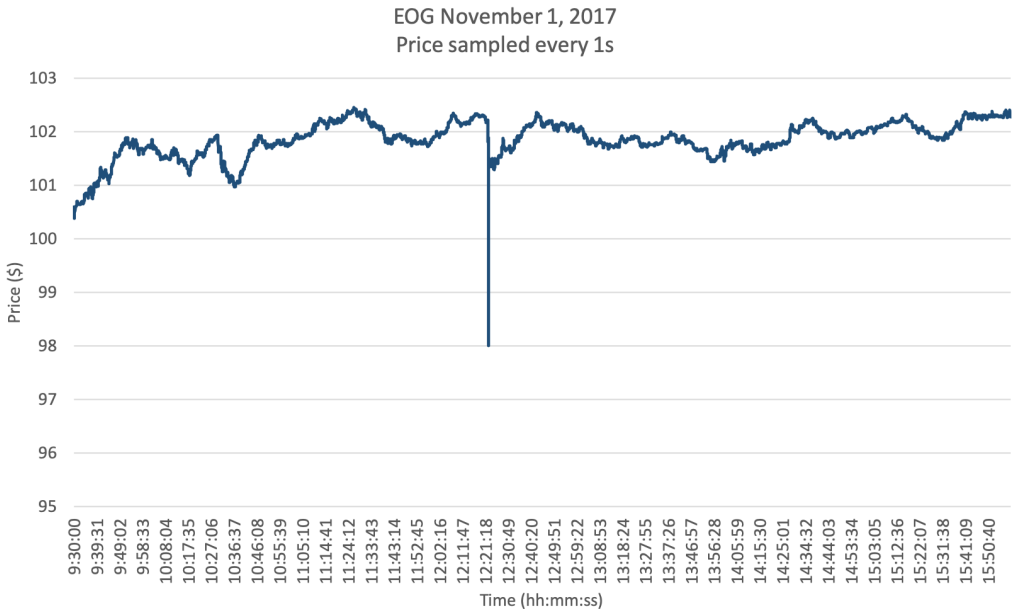


Figure 2.1: The price of EOG Resources, Inc. (symbol EOG) drops dramatically around 12:22:20, but then quickly reverts.

to split potential mini flash crashes into phases and verify that the price dislocates and reverts. I use a two-step process because of the expensive run time needed to sample the price at a 10-millisecond level for all dates and symbols, but splitting mini flash crashes into phases requires a more granular price sampling. Detecting false positives and omitting false negatives is always a possibility, but this two-step approach allows me to detect a dataset in a reasonable amount of time while still verifying price changes with higher granularity, which in conjunction with the exclusion of more illiquid symbols minimizes the likelihood of detecting false positives and omitting false negatives. However, false positives may arise because of news events and other externalities which cannot be captured by the two-step process. False negatives might also occur because of the large time intervals in Detection, which is difficult to avoid with a reasonable runtime.

2.4.1 Detection

The Detection step samples the *NBBO* midpoint at 10-second intervals and searches for four features of mini flash crashes. Figure 2.2 illustrates these features in the previous EOG example. The first feature is a severe price dislocation, which happens around 12:22:20 in the EOG mini flash crash. Second, the price after the dislocation

Algorithm 1 Detection of Mini Flash Crashes

$D :=$ Set of dates
$\Sigma :=$ Set of symbols
for all $d \in D, \sigma \in \Sigma$ **do**
    $flag$.append$\Big($DETECTION$(d, \sigma)\Big)$
end for
for all $(d, \sigma, t) \in flag$ **do**
    $miniFlashCrash$.append$\Big($PHASEIDENTIFICATION$(d, \sigma, t)\Big)$
end for
return $miniFlashCrash$

function DETECTION(d, σ) ▷ Section 2.4.1
    $I :=$ Set of 10-second intervals between 9:30 and 16:00
    for all $i \in I$ **do**
        if $\Big($PRICEDISLOCATIONTEST(d, σ, i) ▷ Section 2.4.1.1
            & LOCALMINTEST(d, σ, i) ▷ Section 2.4.1.2
            & REVERSIONTEST(d, σ, i) ▷ Section 2.4.1.3
            & TRADEVOLUMETEST$(d, \sigma, i)\Big)$ **then** ▷ Section 2.4.1.4
            $t :=$ The end time of interval i
            return (d, σ, t)
        end if
    end for
end function

function PHASE IDENTIFICATION(d, σ, t) ▷ Section 2.4.2
    $t_1 : t - 10$ seconds
    $t_3 : t + 60$ seconds
    $J :=$ Set of 10-millisecond intervals between t_1 and t_3
    $(s, \ell, e) :=$ PHASETRANSITIONPOINTS(d, σ, J) ▷ Section 2.4.2.1
    if $\Big($PRICEDISLOCATIONTESTII(s, ℓ) ▷ Section 2.4.2.2
        & REVERSIONTESTII$(\ell, e)\Big)$ **then** ▷ Section 2.4.2.3
        return (d, σ, s, ℓ, e)
    end if
end function

must be a minimum in a two-hour window to ensure that the mini flash crash is an isolated event during the trading day, rather than simply excess volatility. Third, the price must revert within 30 minutes after the price dislocation to at least 50% of the price immediately preceding the dislocation. Mini flash crashes often revert well within 30 minutes, because these events are shorter in duration than the Flash Crash, where most symbols reverted in about 15 minutes (U.S. Securities and Exchange Commission, 2010). The 30-minute reversion condition ensures that the price does not drop again. If the price does quickly drop again, then an event other than a mini flash crash occurs because this is not a quick event where the price of a symbol stabilizes close to the value preceding the event. The fourth feature is high trading volume during the 10-second interval of the severe price dislocation.

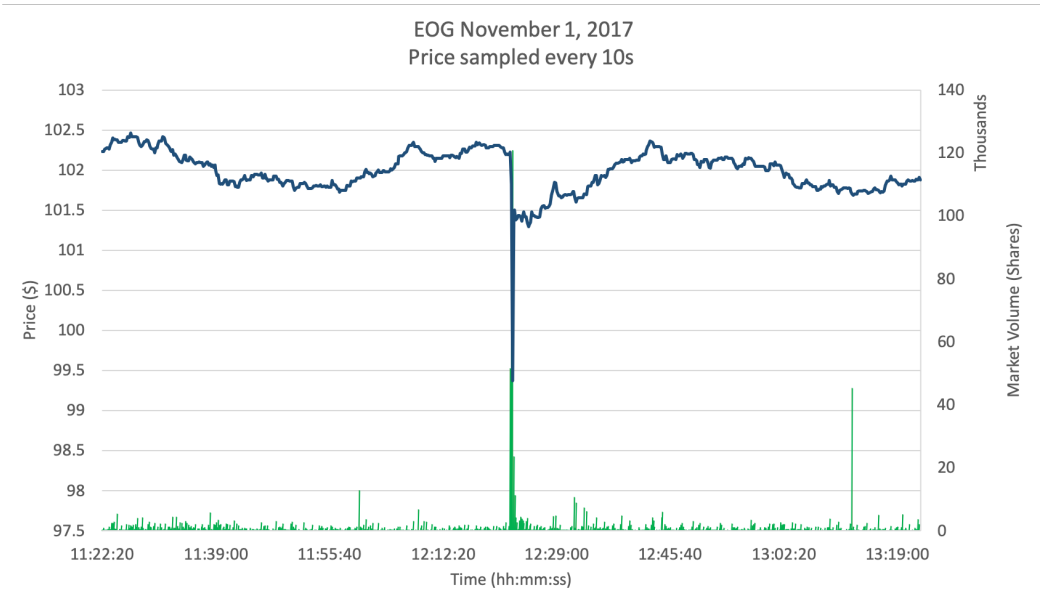


Figure 2.2: Around 12:22:20 the price (blue line) of EOG quickly drops, then reverts quickly afterwards. The total market-wide volume in EOG (green bar) at the drop is significantly higher than any other time. Lastly, the lowest price is a minimum centered in a two-hour window.

2.4.1.1 Price Dislocation

For a 10-second interval i starting at time $t - 10$ seconds and ending at time t, I define the *NBBO* midpoint at the end of interval i as:

$$M_i = \frac{NBB_t + NBO_t}{2}$$

I define *price dislocation* as a percentage during the 10-second interval i as:

$$\delta(i) = 100 \times \frac{M_i - M_{i-1}}{M_{i-1}}$$

An interval i has a severe price dislocation if:

$$\delta(i) < \min\Big\{ -3 \times \overline{|\delta_{5d}|}, -0.3 \Big\} \tag{2.1}$$

where $\overline{|\delta_{5d}|}$ is the average magnitude in price dislocation of all 10-second intervals over the previous five days.[5] The -0.3% threshold here is more relaxed than other studies like Aquilina et al. (2018), Braun et al. (2018), Golub et al. (2017), and Johnson et al. (2013). This relaxed threshold captures many more potential mini flash crashes, but given the infrequent price sampling in Detection, a more lenient threshold reduces the likelihood of missing potential mini flash crashes in the final dataset.

2.4.1.2 Local Minimum

The next criterion is that the price within the interval is the minimum price in the two-hour period centered around the interval in question.[6] If M_i is a local minimum within two-hour period, then this is likely an anomalous event such as a mini flash crash rather than excess volatility. A symbol is at a local minimum if:

$$\min\Big\{M_i, M_{i+1}, M_{i+2}, M_{i+3}\Big\} = \min\Big\{M_{i-360}, M_{i-359}, \ldots, M_i, \ldots, M_{i+359}, M_{i+360}\Big\} \tag{2.2}$$

where there are 360 intervals (equivalent to one hour) on either side of i.

2.4.1.3 Reversion

The price reverts if the average price after the 10-second interval i is close to the average price before interval i. I define the average price between intervals i and j,

[5] Aquilina et al. (2018) also use the average magnitude over a prior date range, but over 20 days rather than 5 days. In my testing, I found that 5 days yields similar results to 20 days.

[6] If interval i is within an hour of the start of the day, the price must be a minimum between the start of the day and an hour after i. If i is within an hour of the end of the day, the price must be a minimum between an hour before i and the end of the day.

where $i \prec j$, as:

$$\overline{M}_{(i,j)} = \frac{1}{(j-i)} \sum_{k:=i}^{j} M_k \tag{2.3}$$

The reversion condition is met if the price after interval i reverts back to a price point greater than or equal to 50% of the price preceding i. This price must reach this price point within 30 minutes.[7] More formally:

$$0.5\Big(\overline{M}_{(i-180,i)} - M_i\Big) \leq \overline{M}_{(i,i+180)} - M_i \tag{2.4}$$

2.4.1.4 Trade Volume

A 10-second interval i must have volume at or above the 95th percentile of all intervals of that day. Trade volume spiked significantly during the Flash Crash (U.S. Securities and Exchange Commission, 2010), so this trade volume condition ensures that a mini flash crash is similar in volume profile to the Flash Crash.

2.4.2 Phase Identification

In the Phase Identification step, I use prices sampled at 10-millisecond intervals to split each potential mini flash crash into four phases. Figure 2.3 depicts the phases of the EOG mini flash crash. The three phase transition points are the time when the crash starts, the time when the crash hits its lowest price, and the time the symbol finishes reverting. After splitting each mini flash crash into phases, I double-check that the price severely dislocates, and reverts at least 50%.

Detection yields a specific symbol and 10-second interval ending at time t for a potential mini flash crash, so Phase Identification examines a smaller time window, $[t_1, t_3]$, where $t_1 = t - 10$ seconds and $t_3 = t + 60$ seconds. Figure 2.4 illustrates the time window examined in Phase Identification. This time window begins ten seconds before t because of the interval length in Detection, but ends sixty seconds after. This was designed in this way because in the Flash Crash, the price in most symbols took longer to revert than drop (U.S. Securities and Exchange Commission, 2010). Increasing the distance between t_1 and t, or between t and t_3, did not yield significantly different results.

[7] If interval i is within 30 minutes of the start of the day, then the average price is taken between the start of day and i. If the interval i is within thirty minutes of the end of the day, then the average

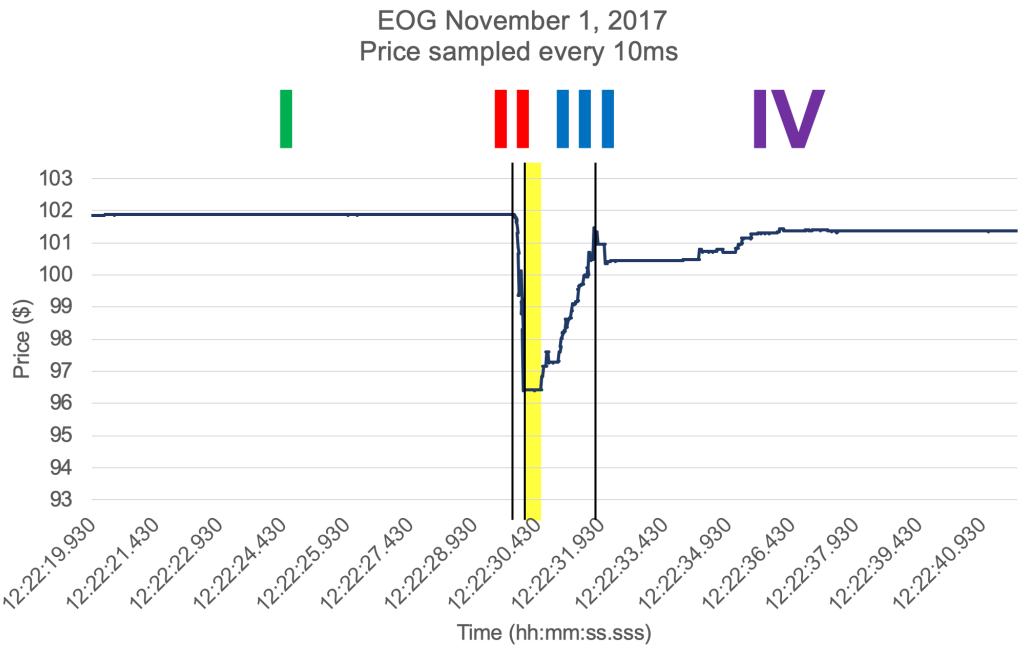


Figure 2.3: The phase division of a EOG mini flash crash with prices sampled at 10-millisecond intervals. Phases I and IV have a 10-second duration before the beginning of the drop and after the end of the reversion, respectively. Phase II is the time between the start of the drop and the time of the lowest price. Phase III is the time between the time of the lowest price and the end of the reversion. EOG's price drops in Phase II, and recovers at least 50% of the drop by the end of Phase III. A LULD threshold was hit between 12:22:30.10 and 12:22:30.53, this time window is highlighted in yellow.

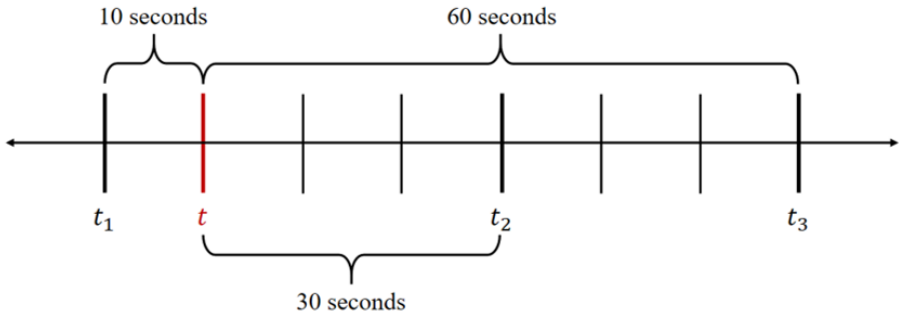


Figure 2.4: Detection yields time t for a potential mini flash crash. Time t_1 is 10 seconds before t, while times t_2 and t_3 are 30 and 60 seconds after t, respectively.

2.4.2.1 Phase Transition Points

For a 10-millisecond interval j starting at time $r - 10$ milliseconds and ending at time r, the *NBBO* midpoint at the end of interval j is:

$$M_j = \frac{NBB_r + NBO_r}{2}$$

I define the phase transition points as the unique start time s, time of the minimum price ℓ, and end time e. The time ℓ of the minimum price of a potential mini flash crash is the end of the 10-millisecond interval j_ℓ, which is defined as:

$$j_\ell = \arg\min \Big\{ M_{j_1}, M_{j_1+1}, \ldots, M_{j_2-1}, M_{j_2} \Big\}$$

where j_1 is the 10-millisecond interval ending at time t_1, and j_2 is the interval ending at time $t_2 = t + 30$ seconds. ℓ is restricted to $\ell \in [t_1, t_2]$, because in my testing, allowing $\ell \in [t_1, t_3]$ tends to yield false positives in more illiquid symbols.

The start s of a mini flash crash is identified by the maximum price between two 10-millisecond intervals j and k, where $j \prec k$. More formally:

$$\max(j, k) = \max \Big\{ M_j, M_{j+1}, \ldots, M_{k-1}, M_k \Big\}$$

The interval j_s where the mini flash crash starts is the last interval preceding j_ℓ for which the maximum price does not increase for at least one second before j_s. More formally:

$$j_s = \begin{cases} \max & j \in \Big\{ j_1, j_1 + 1, \ldots, j_\ell - 2, j_\ell - 1 \Big\}, \\ \text{s.t.} & \forall k \in \Big\{ j - 100, j - 99, \ldots, j - 2, j - 1 \Big\}, \\ & \max(k, j_\ell) = \max(j, j_\ell). \end{cases}$$

Similarly to finding s, the process to find e depends on the maximum price between two 10-millisecond intervals. The interval j_e where the mini flash crash ends is the first interval following j_ℓ for which the maximum price does not increase for at least ten seconds after j_e. More formally:

$$j_e = \begin{cases} \min & j \in \Big\{ j_\ell + 1, j_\ell + 2, \ldots, j_3 - 1, j_3 \Big\}, \\ \text{s.t.} & \forall k \in \Big\{ j + 1, j + 2, \ldots, j + 999, j + 1000 \Big\}, \\ & \max(j_\ell, k) = \max(j_\ell, j). \end{cases}$$

price is taken between i and the end of the day.

The one-second and ten-second thresholds for the amount of time the maximum price cannot increase for s and e, respectively, signifies that the price stabilizes because the price is no longer rapidly changing. The threshold for s is shorter than that for e, because the reversion of a mini flash crash typically takes longer than the drop. The values of these thresholds affect the duration of Phase II and III. The one-second and ten-second thresholds for s and e, respectively, seem to yield more potential mini flash crashes where the price stabilizes before and after the event.

2.4.2.2 Price Dislocation

To verify that a mini flash crash has a severe price dislocation, the threshold is:

$$100 \times \frac{M_{j_\ell} - \overline{M}_{(j_1,j_s)}}{\overline{M}_{(j_1,j_s)}} < -0.7 \tag{2.5}$$

where $\overline{M}_{(j,k)}$ is the average price between intervals j and k defined in Equation 2.3. I tighten the threshold of -0.3% from Detection to -0.7%, because Phase Identification samples the price more frequently. This -0.7% threshold is in line with other studies (Aquilina et al., 2018; Braun et al., 2018; Golub et al., 2017; Johnson et al., 2013).

2.4.2.3 Reversion

A symbol reverts by time e if price recovery between intervals j_ℓ and j_e is at least half of the price drop between intervals j_s and j_ℓ. More formally:

$$0.5\Big(\overline{M}_{(j_1,j_s)} - M_{j_\ell}\Big) \leq \overline{M}_{(j_e,j_3)} - M_{j_\ell} \tag{2.6}$$

The average price preceding interval j_s ensures that the price does not continue increasing more than one second before j_s. The average price following interval j_e confirms that the price does not decrease after j_e.

2.4.3 Detecting Abrupt Price Disruptions

An *abrupt price disruption* is similar to a mini flash crash in every aspect except it reverts 40% or less in Phase Identification. To qualify as an abrupt price disruption, an event must pass checks in Detection, and all except Condition 2.6 in Phase

Identification. Instead I verify that:

$$0.4\Big(\overline{M}_{(j_1,j_s)} - M_{j_\ell}\Big) \geq \overline{M}_{(j_e,j_3)} - M_{j_\ell} \tag{2.7}$$

Abrupt price disruptions still go through four phases. For example, Figure 2.5 shows that the abrupt price disruption in OLED on May 4, 2018 still contains the phase transition points, but did not revert before e.

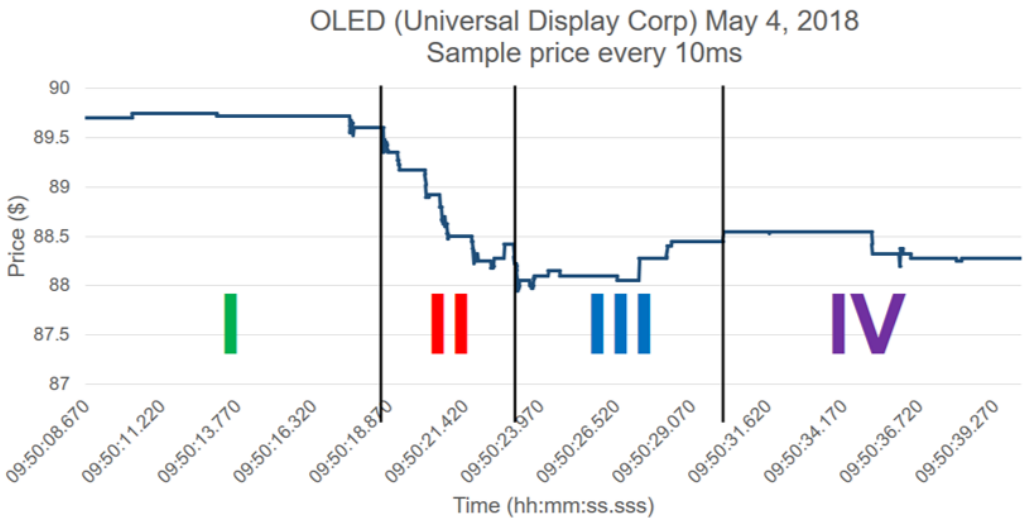


Figure 2.5: The phase division of an OLED abrupt price disruption with prices sampled at 10-millisecond intervals. The phases are similar to those in the example from of Figure 2.3, but the price here does not revert, so Phase III is now the time of the lowest price to the when the price stabilizes.

2.5 Empirical Datasets

I employ my two-step approach described in the previous section to generate a set of potential mini flash crashes and a set of potential abrupt price disruptions, and I study the types of symbols in which these events happen to determine if any patterns exist. The phases of mini flash crashes are variable, so I analyze the duration of each phase. I also examine trading volume across trading venues in potential mini flash crashes to determine if any unusual trends exist on any venue. To determine the potential cost of mini flash crashes to investors, I study the cost in each phase.

2.5.1 Mini Flash Crashes

I detect 632 potential mini flash crashes across 476 distinct symbols over the time period studied.[8] Mini flash crashes appear on 68% of trading days within this thirteen month range. Only 3.64% of mini flash crashes in my dataset represent single name events that occurred during a larger market-wide move, such as during the high market volatility the week of February 5, 2018. Table 2.1 lists some statistics on the frequency and severity of these potential mini flash crashes.

Table 2.1: The number of mini flash crashes per day and per month, the percentage by which the price dislocates, and the price reversion as a percentage of the price dislocation.

	Number per day	**Number per month**	**Dislocation**	**Reversion**
Mean	2.3	49	-1.36%	74.50%
Min	0.0	33	-0.70%	50.02%
Max	13.0	65	-9.96%	184.50%

Tape and market capitalization provide a better look at patterns within the symbols in which mini flash crashes occur. Table 2.2 shows a breakdown of distinct symbols with mini flash crashes by tape, compared to all symbols in June 2018.[9] Table 2.3 shows a breakdown of distinct symbols with mini flash crashes by market capitalization compared to all symbols in June 2018.[10] The distribution of Tape A and mid and large cap symbols in mini flash crashes is generally representative of all symbols. However, there are proportionally more mini flash crashes in Tape C and small cap symbols compared to that of all symbols. This is somewhat expected because more small cap symbols are listed on Tape C than Tape A, and some Tape B and micro cap symbols are excluded because they are less liquid.

The phases vary in length for each mini flash crash. Table 2.4 shows statistics of the time duration of each phase. Phases I and IV are consistently 10 seconds, by

[8] I excluded a potential mini flash crash in the symbol CDXS because the size of a single trade during the suspected event was about 3 times greater than the daily average volume in this symbol.

[9] Tape A symbols are listed on NYSE, Tape C symbols are listed on NASDAQ, and Tape B symbols are all other listings.

[10] Market capitalization or market cap is the total value of all of a company's outstanding shares. A micro cap symbol has a market cap less than $300 million, a small cap symbol has a market cap between $300 million and $2 billion, a mid cap symbol has a market cap between $2 billion and $10 billion, and a large cap symbol has a market cap greater than $10 billion.

Table 2.2: Distribution of distinct symbols by tape in mini flash crashes and all symbols in June 2018.

Tape	Mini flash crash symbols	All symbols
A	32%	37%
B	2%	25%
C	66%	39%

Table 2.3: Distribution of distinct symbols by market capitalization in mini flash crashes and all symbols in June 2018.

Market cap	Mini flash crash symbols	All symbols
Micro	25%	49%
Small	57%	26%
Mid	15%	14%
Large	6%	9%

definition. Phase II tends to be around 1-3 seconds. Phase III is typically closer in length to Phases I and IV.

Table 2.4: Duration in seconds of each phase of a mini flash crash.

	Phase I	Phase II	Phase III	Phase IV
Mean	10.00	2.51	11.71	10.00
Median	10.00	1.03	8.70	10.00
Min	10.00	0.02	0.01	10.00
Max	10.00	31.27	47.27	10.00

Table 2.5 reports the average volume on each venue and the whole US equity market during the phases of a mini flash crash. Most exchanges and the market as a whole follow a Gaussian curve in average volume by phase, with a spike in Phase II. This is particularly prevalent on NSDQ and NYSE, which is likely a result of these exchanges spending more time quoting at the inside of the *NBBO* (Wah et al., 2019). NSX on average reports zero volume in Phase III and IV, possibly because

this exchange spends less time at the inside, so the price reversion typically happens on other exchanges. Both NSX and MKT have a market share less than 2%, which is likely contributes to low trading volume on these exchanges during each phase. Interestingly, volume increased by 37% from Phase II to Phase III on the Trade Reporting Facility (TRF), which is where all off-exchange trades are reported.

Table 2.5: Average volume of shares in each phase of a mini flash crash by trading venue, which includes twelve exchanges, the TRF, and the total US equity market.

Trading Venue	**Phase I**	**Phase II**	**Phase III**	**Phase IV**
Trade Reporting Facility (TRF)	4303	3487	4778	1749
NYSE Arca	895	1205	868	207
NASDAQ BX	445	550	461	136
CBOE BYX Exchange	577	723	594	158
CBOE BZX Exchange	881	1240	787	208
CBOE EDGA Exchange	124	148	150	47
CBOE EDGX Exchange	1466	1380	723	165
Investors Exchange (IEX)	237	259	247	103
NYSE American (MKT)	23	19	8	5
Nasdaq Stock Market (NSDQ)	2700	3793	2436	529
NYSE National (NSX)	2	1	0	0
New York Stock Exchange (NYSE)	1336	2587	1415	316
NASDAQ PSX	103	150	87	19
Total US Equity Market	1007	1196	966	208

I examine the cost of mini flash crashes to investors in *basis points* (bps), which are a hundredth of a percent. Calculating the cost in bps prevents symbol price from skewing results. To determine the potential cost in bps to investors, I compare transaction prices to the *NBBO* midpoint at the start of a mini flash crash. More formally the cost of a single transaction τ in bps is defined as:

$$c_\tau = \begin{cases} \frac{(M_s - p_\tau)q_\tau}{100M_s} & \text{if } p_\tau < M_\tau, \\ 0 & \text{if } p_\tau = M_\tau, \\ \frac{(p_\tau - M_s)q_\tau}{100M_s} & \text{if } p_\tau > M_\tau, \end{cases}$$

where M_s is the *NBBO* midpoint at the start time s of the mini flash crash, M_τ is the *NBBO* midpoint at the time of trade τ, p_τ is the transaction price of trade τ, and q_τ is the quantity of trade τ. I classify a trade as initiated by a sell order if $p_\tau < M_\tau$, and a trade as initiated by a buy order if $p_\tau > M_\tau$. I classified trades as initiated by aggressively priced orders, so these orders cannot be priced to the midpoint. Therefore, calculating the cost based on the midpoint price at the beginning of the event may overestimate the cost to traders, especially while the limit order book is rapidly changing. I define the cost in bps of mini flash crash as:

$$C = \frac{\sum_\tau c_\tau}{\sum_\tau q_\tau} \tag{2.8}$$

Table 2.6 shows the cost in bps of all trades to investors in Phases II, III, and IV; I exclude Phase I because it precedes the start of the mini flash crash. Even though Phase II tends to be the shortest phase, the average cost per mini flash crash in Phase II is at least 10% and 3.75% higher than the average and volume-weighted average cost in other phases, respectively. However, this is somewhat anticipated given the high volume and price drop in Phase II. The cost to investors of Phase III is negative, which is also unsurprising because trades are more likely to be initiated by buy orders when the price reverts. Phase IV also yields a positive cost to investors, which happens because on average mini flash crashes revert 74.5%, so the price does not typically reach the price preceding the event. Overall, mini flash crashes have a negative impact on investors because of the higher positive cost for the majority of the event.

Table 2.6: Estimated cost in bps within each phase to investors. The average cost per mini flash crash and the volumed weighted average cost per mini flash crash.

	Phase II	Phase III	Phase IV
Mean	0.57	-0.004	0.053
Volume-weighted mean	2.23	-1.39	0.59

2.5.2 Abrupt Price Disruptions

I find 365 abrupt price disruptions across 302 distinct symbols over the time period studied. Table 2.7 shows some general statistics on this dataset. As described in Section 2.4.3, abrupt price disruptions can also be split into four phases. Table 2.8 shows the time statistics of each phase. Again, Phases I and IV are constant and 10 seconds in length. Phase II of an abrupt price disruption is typically about 2-5 seconds. Phase III of an abrupt price disruption is similar in length to Phase II, so Phase III of an abrupt price disruption is much shorter than Phase III of a mini flash crash. This is unsurprising because if the price does not revert, then the price likely takes a shorter time to stabilize after the lowest price.

Table 2.7: The number of abrupt price disruptions per day and per month, the percentage by which the price dislocates, and the price reversion as a percentage of the price dislocation.

	Number per day	Number per month	Dislocation	Reversion
Mean	1.3	28	-1.29%	25.70%
Min	0.0	19	-0.70%	0.35%
Max	6.0	42	-9.11%	40.00%

2.6 Market Participant Activity by Phase

I analyze order and trade activity during each phase of potential mini flash crashes using TAQ data. In Phase I, I examine Intermarket Sweep Orders (ISO). Market participants typically use ISOs to quickly trade with shares at multiple price levels, so an ISO that takes out one or more price levels could exacerbate a rapid drop in

Table 2.8: The duration in seconds of each phase of an abrupt price disruption.

	Phase I	Phase II	Phase III	Phase IV
Mean	10.00	4.25	4.84	10.00
Median	10.00	2.02	2.49	10.00
Min	10.00	0.02	0.01	10.00
Max	10.00	31.74	40.72	10.00

price. A limitation of TAQ data is that it does not provide participant and order information, so I propose ideas for future work to analyze market participants who can quickly react to price changes in Phase II. These fast-acting participants could use trend-following strategies to avoid adverse selection, which could potentially intensify the price drop during a mini flash crash. Prior work found that ISOs and HFTs exacerbate mini flash crashes (Aquilina et al., 2018; Braun et al., 2018; Golub et al., 2017). In Phase III, I study off-exchange subpenny trades at prices common for retail volume as a proxy for retail trades to determine if retail investors are affected by mini flash crashes. The analysis of Phase III concludes with a discussion of the limitations of the proxy method used to classify trades initiated by retail orders. Lastly, I look at orders and trades during Phase IV to see if abnormal activity patterns persist after the price stabilizes.

2.6.1 Phase I: Intermarket Sweep Orders on all Exchanges

An ISO is a limit order that can execute on a trading venue even if other venues offer better prices, so an ISO puts the responsibility on the market participant, not the venue, to verify that the order or another one will trade at the best quote across all exchanges. The market participant must verify that an ISO will trade against the best listed quote, then any remaining shares of the ISO can trade against orders on any other exchange. Therefore, a trading venue can execute a trade initiated by an ISO at the venue's best available quote, regardless of the current *NBBO*.[11] A trade initiated by an ISO is marked by an indicator on TAQ data. Prior work by Chakravarty et al. (2012) found that market participants extensively use ISOs and about 46% of trades are initiated by ISOs. There are a variety of reasons a participant might submit a

[11]This is the definition of an ISO from the Securities and Exchange Commission in Regulation National Market System Rule 600(b)(30).

ISO, including exiting a long position or taking out multiple price levels. However, participants can also use ISOs for quicker execution and have no intention of trading with prices inferior to the *NBBO* (Chakravarty et al., 2012). Some participants might regularly use ISOs to immediately execute trades on a certain venue while avoiding a trade-through, which is when a transaction happens at a price worse than the prevailing *NBBO*. For example, if a sell order trades with a buy order at a price below the *NBB*, this constitutes a trade-through because the sell order could have traded at the *NBB*, which is a better price.

I study ISOs during mini flash crashes to determine if they contribute to the initial price drop. A sell ISO could aggravate a mini flash crash, because this order could trade with all shares available both at the *NBB* and at price levels deeper in the book, leading the price to immediately drop. It is possible that a single large ISO could cause a mini flash crash (Braun et al., 2018). It is also possible that multiple participants could exacerbate a mini flash crash with multiple ISOs. If a few participants initiate trades with sell ISOs, and others see the price starting to fall, then these other participants could take out more resting bids with sell ISOs eventually leading to a mini flash crash. Participants potentially submit the initial sell ISOs only to exit a long position. These other participants who submit the subsequent sell ISOs may do so to avoid holding a long position in the case of a price drop and may simply be responding to the price movement.

To determine if an ISO is a buy or sell order, I use a method similar to that of Ellis et al. (2000), who classify trades at the *NBO* as initiated by a buyer and trades at the *NBB* as initiated by a seller. I relax this classification for a mini flash crash because the *NBBO* changes so fast during these events that the exact *NBBO* at the time of the trade is difficult to determine with precision using TAQ data. I classify an ISO execution as initiated by a buy ISO if $p_\tau > M_\tau$, where p_τ is the transaction price of trade τ and M_τ is the *NBBO* midpoint at the time of trade τ. An ISO trade is classified as initiated by a sell ISO if $p_\tau < M_\tau$. Trades where $p_\tau = M_\tau$ are excluded.

To determine if trades initiated by sell ISOs add to the initial drop in mini flash crashes, I analyze ISOs in the time interval beginning one minute before the start of Phase II and ending one minute after the end of Phase III. Figure 2.6 shows the volume initiated by sell and buy ISOs as a percentage of the total market volume during a mini flash crash. The percentage of volume initiated by sell ISOs starts to gradually increase about 30 seconds before the start of Phase II. This is well before the beginning of Phase II, which implies that some market participants may anticipate the

price drop and submit sell ISOs to avoid adverse selection. Therefore, a participant or multiple participants could potentially exacerbate a mini flash crash by initiating trades with sell ISOs to exit a position and sell or short the stock before the price drops further.

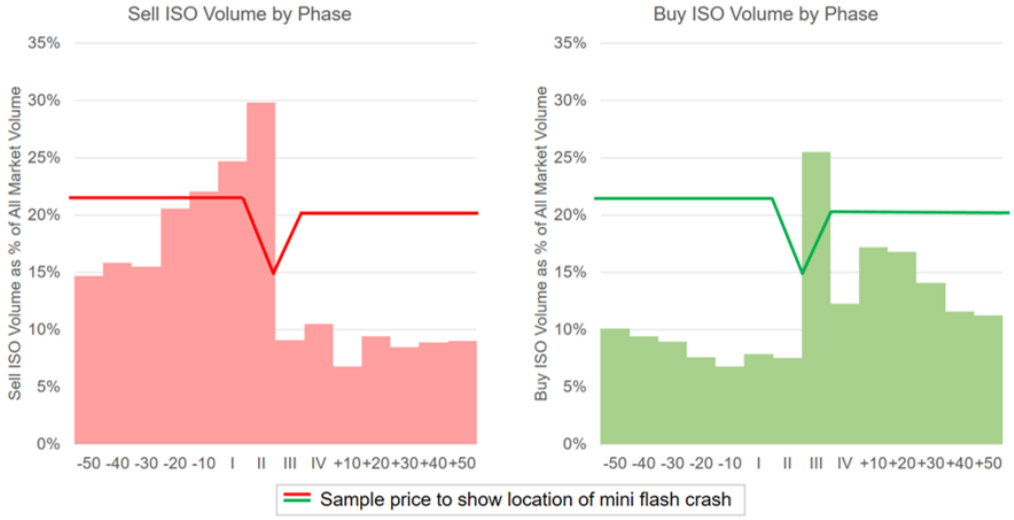


Figure 2.6: The left plot depicts the sell ISO volume as a percent of total volume, while the right plot shows the buy ISO volume as a percent of total volume. The red and green lines on the graphs show the shape of a hypothetical mini flash crash, to demonstrate where the volume is relative to the phase transitions. I study one minute before the start of Phase II and one minute after the end of Phase III, and I split the time preceding and following Phases II and III into 10-second deciles. The point on both plots where the x-axis reads -10 is 10 seconds before Phase I, while +10 is 10 seconds after Phase IV.

I also examine the percentage of volume initiated by sell ISOs in abrupt price disruptions to see if the activity leading into these events and mini flash crashes differs. The pattern of volume initiated by sell ISOs is relatively similar during abrupt price disruptions to mini flash crashes, which suggests that sell ISO activity is independent of later price reversion. The similar sell ISO pattern between these events supports the hypothesis that if multiple trades are initiated by sell ISOs, then other participants might follow the trend to avoid holding a long position in the event of a price dislocation.

2.6.2 Phase II: Potential Order Activity by Participant Type

TAQ data does not provide market participant and order information, but analyzing how certain participants behave during mini flash crashes could potentially provide insight into what exacerbates these events. I propose studying orders by participant type and side (buy, sell, short) during Phase II to determine if certain market participants intensify the price drop more than others. It would be particularly interesting to look at the order activity of *proprietary trading firms* (PTF), who are market participants with no clients and only trades on their own behalf in a principal capacity. PTFs pay for better access to data feeds, and respond to new information faster than other market participants (HFT firms typically fall into this category). PTFs could act as market makers and provide liquidity for other market participants, or could seek to avoid adverse selection by quickly removing liquidity before other market participants. Since some PTFs may have better, more up-to-date market information, they may be able to anticipate the price drop sooner than others. These participants may be employing trend-following strategies to avoid adverse selection. By using superior market information and trend-following strategies, these market participants may be able to quickly avoid adverse selection once they believe the price is about to drop. Submitting marketable orders in the direction of the price drop could potentially intensify mini flash crashes. It is not possible to study orders by participant type with TAQ data, but this proposed work could potentially give a more comprehensive look at how certain market participants contribute to mini flash crashes.

2.6.3 Phase III: Retail Trades on the TRF

In the U.S. equities market, *payment for order flow* is the practice in which brokers purchase investors' orders, then execute these orders against other brokers' orders (Battalio and Holden, 2001; Parlour and Rajan, 2003). *Wholesalers* are brokers who purchase retail investors' orders through payment for order flow, then internalize a portion of these orders. Internalization is when a broker executes an order against its own account. Boehmer et al. (2017) and Parlour and Rajan (2003) observe that for three commonly used wholesalers more than 98% of retail volume through payment for order flow is *undirected*, or does not specify the venue on which an investor wishes their order to transact. It is also suspected that almost 100% of marketable retail order flow is routed through wholesalers (CFA Institute, 2012). In these cases, wholesalers

are free to internalize this volume if they decide to do so. Wholesalers often offer subpenny price improvement on internalized retail orders to attract retail order flow (Levine, 2018). Boehmer et al. (2017) conclude that more than 90% of executed and undirected retail volume receives subpenny price improvement. Retail volume which receives subpenny price improvement is likely initiated by marketable retail orders, because otherwise a retail order will likely be passed to some other trading venue to rest in an order book. An internalized order trades off-exchange, so these trades are reported to the TRF.

To determine if internalized retail volume could potentially contribute in part to the increase in TRF volume in Phase III in Table 2.5, I use TAQ data to study subpenny trades on the TRF as a proxy for retail trades. TAQ data does not indicate when a retail investor is involved in a transaction, so a proxy method must be used to identify retail volume. Subpenny price improvement tends to be on the order of mils, where one mil is one hundredth of a penny. Similar to Boehmer et al. (2017), I identify a trade τ with subpenny price improvement r_τ by:

$$r_\tau = 1000 \times mod(M_\tau, 0.01)$$

where M_τ is the *NBBO* midpoint at the time of trade τ, and $mod(M_\tau, 0.01)$ is the remainder of $M_\tau/0.01$. Figure 2.7 demonstrates if $r_\tau \in [1, 49]$, then trades are potentially initiated by retail sell orders, and if $r_\tau \in [51, 99]$, then trades are potentially initiated by retail buy orders. Retail orders are typically offered price improvement of 1 or 10 mils, so I classify trades where $r_\tau \in [1, 10]$ and $r_\tau \in [90, 99]$ with stronger confidence.[12] More formally, I classify subpenny trades in the following way:

1. If $r_\tau \in [1, 10]$, then this is a strong confidence retail sell trade.
2. If $r_\tau \in [11, 49]$, then this is a weak confidence retail sell trade.
3. If $r_\tau = 50$, then this is a midpoint trade.
4. If $r_\tau \in [51, 89]$, then this is a weak confidence retail buy trade.
5. If $r_\tau \in [90, 99]$, then this is a strong confidence retail buy trade.

[12]Nanex Research found increases in subpenny trades 1 and 10 mils away from the bid or offer, http://www.nanex.net/aqck2/3519.html. They classify these trades as subpenny price-improved retail orders. Blankespoor et al. (2018) and Boehmer et al. (2017) also find that 1, 10, and 20 mils of price improvement are most common for retail investors.

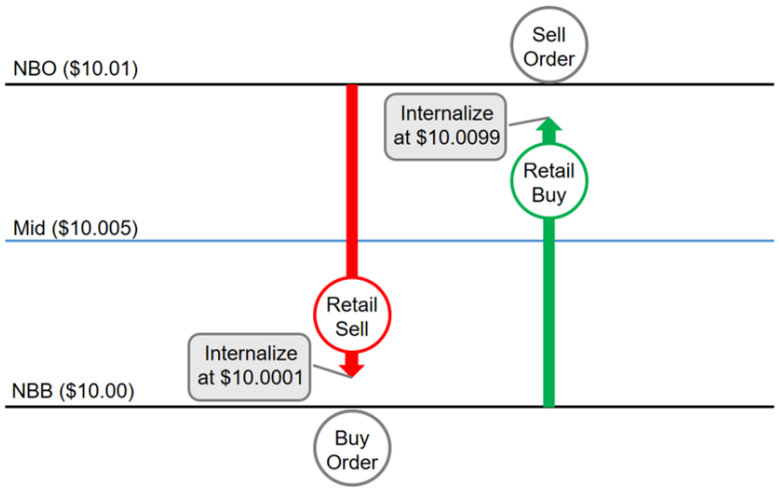


Figure 2.7: An example of how subpenny trades are classified as retail. The *NBB* is \$10.00 and the *NBO* is \$10.01. A retail order crossing the spread might be internalized by a wholesaler who offers subpenny price improvement. In such cases a retail sell order would execute at a price a subpenny increment above \$10. A retail buy order would execute at a price a subpenny increment below \$10.01.

Trades where $r_\tau = 50$ are classified as midpoint trades, because the minimum pricing increment is \$0.01. Off-exchange trades are not identified as being initiated by a retail investor, so non-retail trades may be captured in this classification method. This method most likely does not capture all retail volume, since only about 88% of executed retail volume receives subpenny price improvement (Boehmer et al., 2017).

I examine subpenny volume on the TRF as a percentage of total TRF volume during each phase of a mini flash crash to determine if trades potentially initiated by retail investors contribute to the increase in volume in Phase III. Figure 2.8 shows retail volume on the TRF as a percentage of the total TRF volume. A higher volume is potentially initiated by retail sell orders in Phase I and II, whereas a higher volume is potentially initiated by retail buy orders in Phase III. Strong confidence retail volume makes up about 11% of volume on the TRF during Phase III, suggesting that retail investors are potentially affected by these events.

To determine if mini flash crashes impact retail investors, I study strong confidence retail volume. Since retail investors are more likely to receive price improvement of 1 and 10 mils, I isolate trades where $r_\tau \in \{1, 10, 90, 99\}$. Figure 2.9 depicts retail volume in each phase, where the time within each phase is divided into deciles. Surprisingly, there is a large increase in potential retail sell volume in the first decile of Phase III,

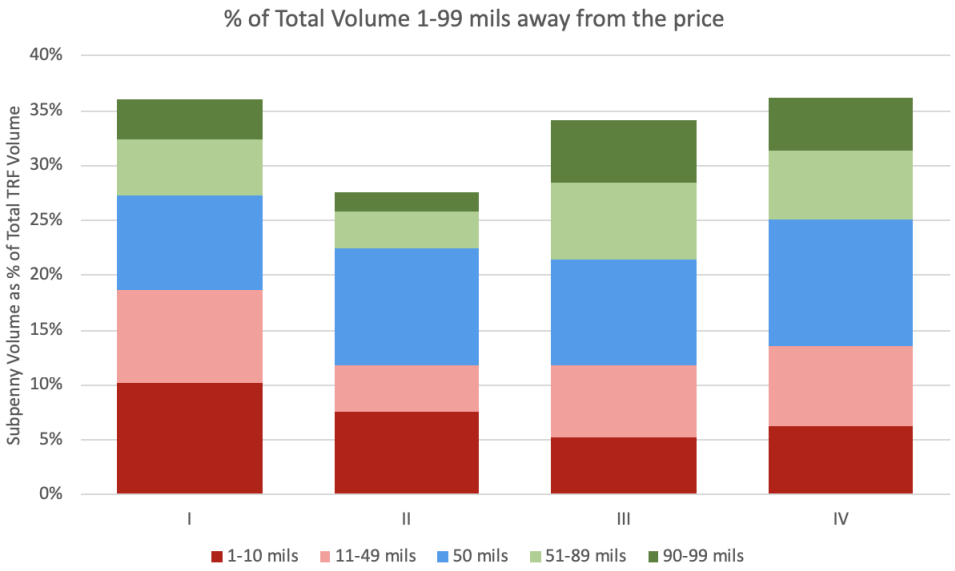


Figure 2.8: Subpenny volume on the TRF as a percentage of all TRF volume in each phase.

where the retail sell volume increases by 278% from the average retail sell volume across the deciles of Phase II. The majority of potential retail sell volume occurs around the lowest price most likely because retail orders submitted in response to mini flash crashes may transact later than orders from other participants. Retail investors may not react as quickly as other participants because they typically do not pay for access to better data feeds and do not possess technology optimized to trade as quickly as possible. Retail investors also use *stop-limit orders*, which are orders that automatically trade when a symbol reaches a certain price. A stop-limit sell order transacts when the price drops to a certain level, so this order type most likely will transact at a lower price around the price drop. Therefore, the observed spike in potential retail sell volume in the first decile of Phase III is potentially because of their slower response times and the use of stop-limit orders.

I also analyze strong confidence retail volume during abrupt price disruptions. Figure 2.10 shows the retail volume of each phase of an abrupt price disruption, where $r_\tau \in \{1, 10, 90, 99\}$ and each phase is divided into deciles. Potential retail volume in the first decile of Phase II is more than five times the average retail volume across all deciles of an abrupt price disruption. Interestingly, the potential retail sell volume during a mini flash crash peaks around the lowest price, whereas the potential retail sell volume during an abrupt price disruption peaks around the price just before the

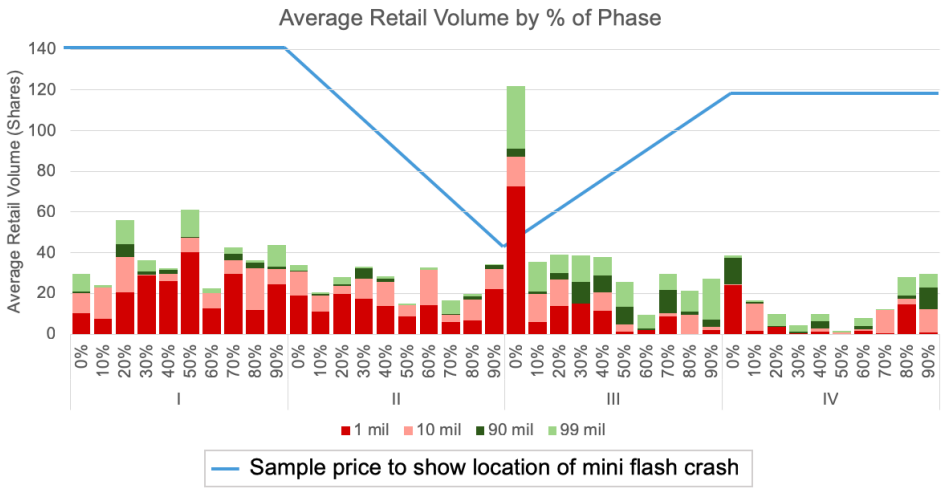


Figure 2.9: Retail volume in each phase of a mini flash crash, where 1 and 10 mils are trades initiated by retail sell orders and 90 and 99 mils are trades initiated by retail buy orders. Each phase is split into deciles. The blue line shows the shape of a hypothetical mini flash crash, to demonstrate where the volume is relative to the phase transitions.

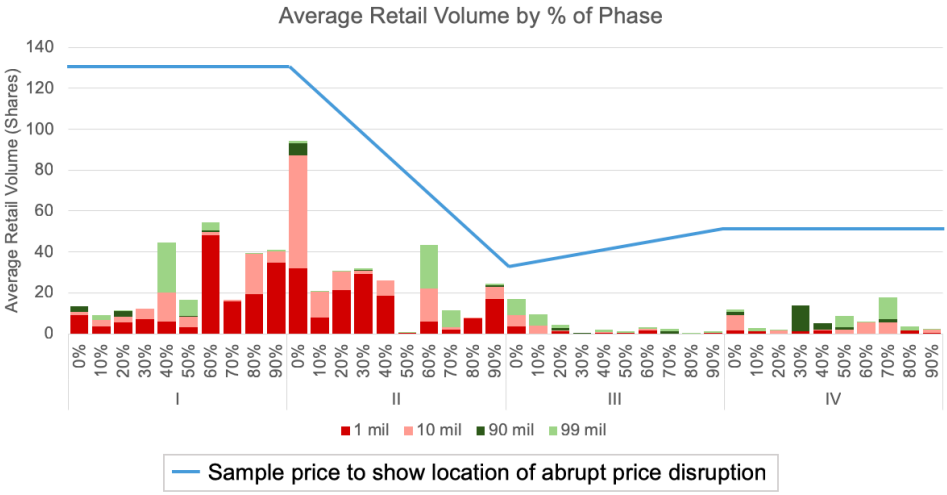


Figure 2.10: Retail volume in each phase of an abrupt price disruption, where 1 and 10 mils are trades initiated by retail sell orders and 90 and 99 mils are trades initiated by retail buy orders. Each phase is split into deciles. The blue line shows the shape of a hypothetical abrupt price disruption, to demonstrate where the volume is relative to the phase transitions.

drop. The total retail volume (over all four phases) of abrupt price disruptions is just above half the total retail volume (over all four phases) of mini flash crashes. One potential explanation for this is that fewer retail investors trade in the symbols in which I observed potential abrupt price disruptions, so there is lower retail volume overall. Another explanation is that instead of internalizing, wholesalers may be passing more orders onto exchanges during abrupt price disruptions. Retail volume might be passed to exchanges during abrupt price disruptions because if retail volume exists in mini flash crashes, it potentially also exists in abrupt price disruptions. This is especially true for stop-limit orders, because these orders will enter the market as long as a symbol drops to the specified limit-price. Therefore, if stop-limit order prices are hit during mini flash crashes, then these prices are also likely hit in abrupt price disruptions. However, there appears to be significantly less retail volume in abrupt price disruptions, which is why retail volume is potentially passed to exchanges during these events.

Retail investors are potentially negatively impacted by mini flash crashes, because retail sell orders submitted during these events may be more likely to execute at worse prices than anticipated. Stop-limit orders, which are often used by retail investors and which do not transact until the price reaches a certain level, may be more likely to be triggered during these events. I assess the potential cost to retail investors by comparing the actual transaction price to the price at the start of the mini flash crash. Table 2.9 shows the estimated cost of each phase for trades where $r_\tau \in \{1, 10, 90, 99\}$, which is at least 10% and 3.5% lower than the average cost and volume-weighted average cost of all trades in Phase II and IV, respectively (Table 2.6). In other words, the cost to retail investors is only a small portion of the cost to all investors. My estimates suggest that there is nonzero cost to potential retail investors in Phase III, which is largely due to the spike observed in the first decile of Phase III. This is only one way to estimate the cost to retail investors, and there are many other possible metrics to calculate cost. Larger volatility events like the Flash Crash likely pose greater costs to all investors, not just retail investors.

2.6.3.1 Limitations of Retail Proxy Method

As discussed at the beginning of Section 2.6.3, a high proportion of retail volume is potentially internalized at subpenny prices, but it is difficult to know the exact percent of retail orders captured by this method. It is likely that this classification

Table 2.9: The cost in bps of each phase to retail investors. The average cost per mini flash crash and the volume weighted average cost per mini flash crash.

	Phase II	Phase III	Phase IV
Mean	0.03	0.01	0.002
Volume-weighted mean	0.62	0.24	0.09

method excludes a portion of retail orders. Any retail volume that is not executed, internalized by wholesalers, or offered subpenny price improvement is not captured by this classification method because this volume will not trade off-exchange at subpenny prices.

Along with potentially misclassifying retail volume with this proxy method, it is also possible that this method misclassifies non-retail volume as retail. Wholesalers internalizing retail orders are potentially not the only market participants internalizing orders at subpenny prices. Large broker-dealers, such as large institutional banks, generally internalize a lot of order flow. While it is possible that these internalized trades could execute at subpenny prices, there is little to no evidence that this is a frequent occurrence. The biggest concern is that this method captures internalized order flow from *single-dealer platforms* (SDP), which receive and then internalize or reject orders from institutional investors (Ye, 2016). There is very little transparency in SDPs, so it is unclear how they execute trades and what kind of subpenny price improvement they potentially offer (Massa, 2018). It is possible that this classification method captures some trades internalized by SDPs.

Trades initiated by retail orders may be misclassified based on this proxy method. It is possible that retail orders are offered price improvement at values other than 1 and 10 mils, and these other values would be excluded when the set of retail volume is further refined to where $r_\tau \in \{1, 10, 90, 99\}$. However, another study concluded that it is unlikely for price improvement to be offered at other values (Boehmer et al., 2017). Also, the proportion of volume at 1 and 10 mils captured by this classification method is fairly close to a study done by Nanex Research (2012). It is also possible that this method misclassifies the side initiating the retail trade. For a sell (buy) to be classified as a buy (sell) the wholesaler must offer 51 mils of price improvement or more, and for a sell (buy) to be classified as a strong confidence buy (sell) the wholesaler must offer 90 or 99 mils of price improvement. It is possible to misclassify the side using my

method, but it is unlikely that a wholesaler would offer this much price improvement.

Considering the slower speed of retail investors, it would be difficult for these investors to actively participate in an event as short in duration as a mini flash crash. It is not possible to be fully confident in the order type of retail volume without attributed order-level data. However, given that mini flash crashes generally happen faster than most retail investors can react, retail volume during a mini flash crash is likely largely initiated by stop-limit orders, which are likely marketable during price dislocations by design. Also, the U.S. Securities and Exchange Commission (2010) found that the majority of the retail orders submitted during the Flash Crash were stop-limit orders, which further supports this hypothesis. Considering that the spike in retail sell volume is observed in the first decile of Phase III, the retail orders that initiated by these trades were likely stop-limit orders submitted during Phase II.

It is also worth considering why wholesalers potentially internalize more orders in the first decile of Phase III. Wholesalers can either internalize retail order flow or pass it to other trading venues. Within the requirements of Reg NMS Rule 611, wholesalers have up to a second to execute a retail order before they commit a trade-through. Table 2.4 shows that the median price drop of a mini flash crash is about one second long. A retail investor may submit a stop-limit order during the price drop, but the price might change direction within the one-second time window for execution. Wholesalers take short-term risks when internalizing retail order flow (CFA Institute, 2012), so in the first decile of Phase III a wholesaler could potentially recognize a change in the price direction and decide to internalize the retail sell volume. Wholesalers may not know that the price is reverting, but rather take on short-term risk that the price movement changes direction at the bottom of these events.

There may also be errors in the *NBBO* timestamps because of the rapid price changes and short duration of mini flash crashes. The *NBBO* timestamps are possibly inaccurate because the quick duration and high activity during mini flash crashes which may lead to stale quotes listed during these events. Therefore, orders may potentially be trading at the end of Phase II rather than the beginning of Phase III.

The difference in potential retail volume in mini flash crashes and abrupt price disruptions might occur because of anomalies in the abrupt price disruption dataset and the duration of abrupt price disruptions. There is a large spike in volume in the first decile of Phase II of abrupt price disruptions compared to mini flash crashes, but it is unlikely that a wholesaler could distinguish between these events at the beginning of the price drop. Given this and that the abrupt price disruption dataset is

about half the size of the mini flash crash dataset, the spike in volume during abrupt price disruptions could potentially be the result of an outlier with abnormal trading behavior in the dataset. There is less incentive for wholesalers to internalize retail sell orders at the lowest price of an abrupt price disruption because there is not necessarily a change in the direction of price movement in the short-term, so it is less likely to be profitable for wholesalers to internalize these orders. Also, Table 2.8 shows that the median time of an abrupt price disruption price drop is about two seconds, so the price might still be moving down one second after a wholesaler receives a retail order. If a wholesaler receives an order more than one second before end of the price drop, then there is again no potential profit by internalizing the order because of a change in price direction in the short-term.

2.6.4 Phase IV

To determine if unusual activity continues after the price stabilizes and finishes reverting, I study trade and order activity in Phase IV. Table 2.5 shows that volume decreases on every trading venue in Phase IV. Therefore, any unusual trade activity is not frequent enough to distinguish any pattern across the set of potential mini flash crashes. Overall, trade activity in Phase IV of mini flash crashes and abrupt price disruptions shows no noteworthy patterns.

2.7 Conclusion

In this chapter, I studied mini flash crashes, which are frequent, miniature renditions of the 2010 Flash Crash, to provide insight into market activity during larger volatility events. Using a two-step process, I generated a dataset of potential mini flash crashes to study order and trade activity during these events. To facilitate a more dynamic and granular comparison of activity patterns during different stages of mini flash crashes, I divided each event into phases, then analyzed trades using TAQ data.

To determine if any market participants exacerbate mini flash crashes, I analyzed different participant activity during each phase, and found that more notable trade activity happens around the phase transition points. I discovered that the proportion of sell ISOs gradually increased before the initial drop in price. This implies that some participants may anticipate the price drop and submit sell ISOs, which can potentially exacerbate a mini flash crash.

I studied off-exchange subpenny trades at 1 and 10 mils as a proxy for retail trades. I discovered that the volume of potential retail sells at or near the lowest price increased by 278% from the average volume of potential retail sells throughout the price drop. Stop-limit orders may be a high proportion of retail volume during mini flash crashes because retail investors are typically not fast enough to actively trade during these events. Stop-limit orders are more likely to trade at lower prices during a mini flash crash because these orders do not transact until the price drops to a certain value. I also found a very low volume of trades initiated by retail investors near the lowest price of abrupt price disruptions. This suggests that retail investors are potentially negatively impacted by mini flash crashes because retail sell volume may be more likely to be internalized at or near the lowest price of these events. It is unlikely that wholesalers can distinguish between a mini flash crash and an abrupt price disruption at the lowest price of these events, so the spike in potential retail sell volume at the lowest price of mini flash crashes is probably the result of wholesalers realizing a potential profit because of a change in the direction of price movement in the short-term.

One limitation of this work is that I focused on events that experience steep and sudden drops in price, but a natural next step would be to expand this dataset to include events that increase in price. Another limitation is that I did not consider external events that could trigger mini flash crashes: incorporating news events and overall market volume and volatility in an analysis of mini flash crashes would provide additional insight into causes of these events and the resulting trading activity. It would be useful to examine how many mini flash crashes reached LULD thresholds, and if any events were eliminated in the second, more granular step of detecting these events because a LULD trading halt was placed on a symbol. It would also be informative to test the results of this chapter in a simulated environment to better analyze market activity by connecting participant information, orders, and trades on the whole market. Simulation would also allow control of the influence of externalities. Another potentially interesting vein of work would be to study the pattern of ISOs across different venues during mini flash crashes to determine if these events are potentially more likely if ISOs are submitted to multiple venues. Given that TAQ data does not provide market participant or order information, it would also be useful to perform the proposed work in Section 2.6.2 on order activity by participant type during Phase II. Considering the spike in potential retail volume in the first decile of Phase III of mini flash crashes, examining the potential retail volume on mini flash

crashes and abrupt price disruptions when the price drop is one second or less, might provide noteworthy insight into why this volume spike occurs. Lastly, retail investors tend to trade in some symbols more than others, so analyzing retail activity on heavily traded retail symbols may provide more insight into the impact on these investors during mini flash crashes.

The results found in this chapter can provide intuition about what market participant activity escalates larger volatility events like the Flash Crash. A high volume of trades initiated by sell ISOs may intensify an event larger than a mini flash crash. Retail investors may also be subject to more harm in larger volatility events than smaller events. If these strategies are used in conjunction by a single participant, then that participant may be intentionally triggering a subset of these events on symbols heavily traded by retail investors. Retail investors may be more vulnerable to these practices now because of the recent increase in retail volume and market volatility. In the case of a larger event this could potentially cause significant harm to the market. Reducing the extent to which this type of activity happens in the U.S. equities market may reduce systemic risk by mitigating the likelihood of larger volatility events in the future.

CHAPTER III

Agent-Based Financial Market Model

The remainder of this thesis utilizes agent-based models of financial markets to study market responses to various trading strategies. In Chapters V and VI, I use a market model developed at the University of Michigan, *Market-Sim* (Wellman and Wah, 2017). I use another agent-based market model, *ABIDES*, developed by Byrd et al. (2019) in Chapter IV. These models share many characteristics, including their structure, trading agents, and compatibility with *Empirical Game-Theoretic Analysis* (EGTA).

Market-Sim and ABIDES support a *continuous double auction* (CDA) with discrete time steps and a finite time horizon T. Orders submitted to a CDA can match with other orders immediately upon arrival. A stock exchange is a common example of an existing CDA. The orders submitted to our CDA are *limit-orders*, which are orders with a limit price attached. For a buy limit-order, the limit price specifies the maximum price at which the order will match with a sell order. For a sell limit-order, the limit price denotes the minimum price at which the order will match with a buy order. Any limit-order which does not immediately match with another order is placed in the *order book*. An order book is a compilation of all unmatched orders organized into two sides, buy and sell. Each side of the book is ordered by price and time priority. A higher priced buy order or lower priced sell order have a higher probability of matching with any competitively priced incoming orders. If two orders have the same limit price, then whichever arrived to the market first gets matching priority.

I use these market models to study the interactions between agents and the market response to certain trading strategies. An agent-based model simplifies financial markets to the point that it can be difficult to extrapolate the results to realistic settings, but historical data is unresponsive to strategies which are not implemented

in the market. I focus on agent-based modeling to examine how other agents in the market react to strategies that leverage market information and microstructure. Agent-based modeling makes it possible to compare the difference in profit between market scenarios when these strategies are deployed and when they are not. I use EGTA to strengthen the findings in these experiments. EGTA approximates equilibria in complex games. Studying trading strategy sets found with EGTA ensures that all agents are playing the optimal strategy in each scenario.

In Section 3.1 I discuss the background agents' valuation model of the traded assets. I explain the background trading strategies used throughout Market-Sim and ABIDES in Section 3.2. In Section 3.3 I provide an overview of how I incorporate agents which generate strategies using deep reinforcement learning (DRL) into Market-Sim. Lastly, I provide an overview of EGTA in Section 3.4.

3.1 Valuation Model

A trading agent in the market evaluates and determines the value of the asset before submitting an order. An agent's valuation of the asset is composed of two parts, a common fundamental value and a private value. The common *fundamental value* is shared by all agents in the market, and represents a shared belief in the assets true value.

I denote the fundamental value of the underlying security at time t by r_t. This fundamental, r_t, varies throughout the simulation by a stochastic mean-reverting process:

$$r_t = \max\{0, \kappa\bar{r} + (1-\kappa)r_{t-1} + u_t\}, t \in [0, T]; r_0 = \bar{r}, \tag{3.1}$$

where $\kappa \in [0, 1]$ specifies the degree to which the time series reverts back to the fundamental mean $\bar{r}$, and $u_t \sim \mathcal{N}(0, \sigma_s^2)$ is the random shock at time t.

ABIDES introduces a series of *mega-shocks* throughout the trading period. These mega-shocks M_t arrive according to a Poisson process and apply price deviations drawn from a bimodal normal distribution with positive and negative modes substantially away from zero. Both sub-distributions are normal distribution $M_t \sim \mathcal{N}(0, \sigma_m^2)$. The inclusion of mega-shocks is used as a method to represent exogenous events which change the perceived value of the asset.

Upon each arrival, an agent updates its estimate of the fundamental value $\hat{r}_t$ at

the current time step t from their last arrival at time step t'. Noise drawn from a normal distribution, $n_t \sim \mathcal{N}(0, \sigma_n^2)$, is added to each agent's fundamental estimate: $o_t = \hat{r}_t + n_t$ (Wang and Wellman, 2017).

3.1.1 Agents' Fundamental Value Estimate in Market-Sim

Given a new noisy observation o_t, an agent estimates the current fundamental by updating its Bayesian posterior mean $\tilde{r}_t$ and variance $\tilde{\sigma}_t^2$. The agent must also update the previous posterior distribution from their last arrival at t'. The previous posterior mean and variance take into account the fundamental mean reversion during the time between the current and last arrival, $\delta = t - t'$. The agent does not update its estimate of the fundamental at the time steps between t' and t, so an agent first updates its estimates of the prior posterior distribution by applying mean reversion. The updated previous posterior mean and variance are given by:

$$\tilde{r}_{t'} \leftarrow (1 - (1-\kappa)^{\delta})\bar{r} + (1-\kappa)^{\delta}\tilde{r}_{t'};$$

$$\tilde{\sigma}_{t'}^2 \leftarrow (1-\kappa)^{2\delta}\tilde{\sigma}_{t'}^2 + \frac{1-(1-\kappa)^{2\delta}}{1-(1-\kappa)^2}\sigma_s^2.$$

The update equations for the previous posterior distribution are effectively weighted distributions between $\bar{r}$ and $\tilde{r}_{t'}$ for the mean and σ_s^2 and $\tilde{\sigma}_{t'}^2$ for the variance. As the time between arrivals δ increases, more weight is placed on the population statistics $\bar{r}$ and σ_s^2 and less weight is placed on the previous estimates $\tilde{r}_{t'}$ and $\tilde{\sigma}_{t'}^2$. Once the agent calculates the previous posterior distribution, it can calculate the posterior distribution at the current time step t:

$$\tilde{r}_t = \frac{\sigma_n^2}{\sigma_n^2 + \tilde{\sigma}_{t'}^2}\tilde{r}_{t'} + \frac{\tilde{\sigma}_{t'}^2}{\sigma_n^2 + \tilde{\sigma}_{t'}^2}o_t; \tag{3.2}$$

$$\tilde{\sigma}_t^2 = \frac{\sigma_n^2\tilde{\sigma}_{t'}^2}{\sigma_n^2 + \tilde{\sigma}_{t'}^2}. \tag{3.3}$$

With the posterior mean, an agent can calculate an estimate $\hat{r}_t$ at time t of the terminal fundamental r_T:

$$\hat{r}_t = \left(1 - (1-\kappa)^{T-t}\right)\bar{r} + (1-\kappa)^{T-t}\tilde{r}_t. \tag{3.4}$$

3.1.2 Agents' Fundamental Value Estimate in ABIDES

Agents in ABIDES use a sparse discrete fundamental value estimate, first proposed by Chakraborty and Kearns (2011). Given an estimate of the fundamental $\hat{r}_{t'}$ from the previous arrival, an estimate of the current fundamental value $\hat{r}_t$ is obtained through an Ornstein–Uhlenbeck process by sampling from a normal distribution $\hat{r}_t \sim \mathcal{N}(\tilde{r}_t, \tilde{\sigma}_n^2)$, where:

$$\tilde{r}_t = \bar{r} + (\hat{r}_{t'} - \bar{r})e^{-\kappa t}; \tag{3.5}$$

$$\tilde{\sigma}_t^2 = \frac{\sigma_s^2}{2\kappa}(1 - e^{-2\kappa t}). \tag{3.6}$$

There exists an implementation difference for the existing agents' valuation in the two market simulation platforms. The agents trading on the Market-Sim platform first estimate the current fundamental, then use that to estimate the final fundamental value. The agents trading on the ABIDES platform solely estimate the current fundamental value. Thus, the existing agents in Market-Sim use an estimate of the final fundamental in their valuation, while the existing agents in ABIDES use an estimate of the current fundamental in their valuation function.

3.1.3 Agents' Private Values

The private value component of an agent's valuation, represents any additional value the asset has to the agent and differs between each trader. A private value vector Θ_i captures the position preference of a trader i. The vector Θ_i has length $2q_{\max}$, where $q_{\max}$ is the maximum number of units an agent can be long or short at any time. Element θ^i_{q+1} represents the marginal gain from buying an additional unit given the current net position q. I produce Θ_i from a set of $2q_{\max}$ values independently drawn from $\mathcal{N}(0, \sigma^2_{PV})$. Next, I sort the elements in Θ_i in order of diminishing marginal utility, so that $\theta^{q'} > \theta^q$, for all $q' < q$. An agent's valuation for a unit of the security at time t is the sum of its private value at the current position q_t and an estimate at time t of the final value of the fundamental, $\hat{r}_t$, more formally:

$$v_i(t) = \begin{cases} \hat{r}_t + \theta_i^{q+1} & \text{buying,} \\ \hat{r}_t - \theta_i^{q} & \text{selling.} \end{cases} \tag{3.7}$$

A trader's final market surplus is the final valuation of its holdings at time T. The final valuation of background trader i with final holdings H and cash flow c:

$$V_i(T) = \begin{cases} r_T H + c + \sum_{k=1}^{k=H} \theta_i^k & \text{long positions } H > 0, \\ r_T H + c - \sum_{k=H+1}^{k=0} \theta_i^k & \text{short positions } H < 0. \end{cases} \tag{3.8}$$

3.2 Background Trading Strategies

A variety of background trading strategies are deployed in Market-Sim and ABIDES. An agent using an impact strategy, defined in Section 3.2.3, arrives to the market at a specified time. Agents using any other trading strategy arrive to the market by a Poisson process with rate λ. When an agent arrives at the market, it withdraws any previous orders under the assumption that any prior market information is now stale.

3.2.1 Zero Intelligence

All of the market environments explored in the next three chapters contain background agents that use a *zero intelligence* (ZI) trading strategy (Gode and Sunder, 1993; Farmer et al., 2005). With equal probability, a ZI agent submits a new limit-order to buy or sell. When submitting an order, a ZI agent i selects a limit-order price by determining a desired additional surplus drawn from a uniform distribution $\zeta_t \sim \mathcal{U}[R_{\min}, R_{\max}]$. The agent then offsets its valuation of the asset $v_i(t)$ at time t by the additional surplus to determine the limit-order price:

$$p_i^{\mathbf{ZI}}(t) \sim \begin{cases} \mathcal{U}\Big[v_i(t) - R_{\max}, v_i(t) - R_{\min}\Big] & \text{buying}, \\ \mathcal{U}\Big[v_i(t) + R_{\min}, v_i(t) + R_{\max}\Big] & \text{selling}. \end{cases}$$

If a ZI agent is buying, the best available sell order ASK_t may be such that:

$$\text{ASK}_t \in \Big(v_i(t) - \zeta_t, v_i(t) - \zeta_t \eta], \tag{3.9}$$

where $\eta \in [0, 1]$. If a ZI agent is selling, the best available buy order BID_t may be such that:

$$\text{BID}_t \in \Big[v_i(t) + \zeta_t \eta, v_i(t) + \zeta_t\Big). \tag{3.10}$$

In either of these scenarios it may be better for the agent to forgo a fraction of its

desired surplus to guarantee an immediate trade. Thus, a ZI trader utilizes a strategic surplus threshold parameter η to consider the current visible quoted price. If a ZI agent could gain a fraction η of its desired surplus by accepting the most competitive visible order, then it will take that quote by submitting a limit-order at the same price.

3.2.2 ZI with Transaction Updates

In Market-Sim and ABIDES, I employ an extension of the ZI strategy (ZIT) developed by Brinkman (2018). When submitting an order, the standard ZI strategy considers its valuation of the asset, a desired surplus, and the current market price. Given that an agent observes noisy estimates of the fundamental, it could learn more information about the fundamental from considering other agents' observations. The observations of other agents are reflected in transactions, so a ZIT agent considers market transactions as evidence of other agents' beliefs to improve its final fundamental estimate. The ZIT strategy handles the price of a previous transaction at time t as an observation drawn from a normal distribution centered around the fundamental value at time t. In Market-Sim, whenever a ZIT agent observes a transaction it updates the posterior distribution of the fundamental with Equations 3.2 and 3.3. In ABIDES, a ZIT agent updates fundamental distribution with Equations 3.5 and 3.6. However, rather than using a noisy observation of the fundamental o_t, the agent uses the transaction price at time t. The agent also uses a transaction observation variance $\sigma_\rho^2 \in [0, \infty)$ in lieu of the observation variance σ_n^2. When transaction observation variance is infinite, a ZIT agent ignores previous transaction, making it equivalent to a ZI agent.

3.2.3 Impact Agent

Chapter IV implements an impact agent to induce a mini flash crash. Beginning at time τ, this agent submits a rapid series of large marketable sell orders. This method consumes the bid side of the order book, causing the price to rapidly drop. Once other traders continue to submit orders, the price typically recovers.

3.2.4 Fundamental Market Maker

I use a market making agent adapted from Wah et al. (2017) in Chapters V and VI. A market maker supplies liquidity to facilitate trading in the market. When a

market maker enters the market at time t, it submits a *ladder* of buy and sell orders around the estimated fundamental value $\hat{r}_t$. There are n rungs on each side of the ladder, or n buy orders and n sell orders. A spread S separates the high priced buy order and lowest priced sell order, and ξ ticks separate each rung on each side of the ladder. More formally, the order ladder a market maker submits at time t:

$$\begin{cases} \hat{r}_t - \frac{S}{2} - \xi i, \text{ s.t. } i \in \{0, 1 \ldots, n-1\} & \text{buy orders,} \\ \hat{r}_t + \frac{S}{2} + \xi i, \text{ s.t. } i \in \{0, 1 \ldots, n-1\} & \text{sell orders.} \end{cases}$$

3.3 Implementing Deep Reinforcement Trading Strategies

In Chapters V and VI, I present work where a trading agent generates its strategies through DRL. I only execute this type of agent in Market-Sim. Market-Sim is written in Java, but the large majority of machine learning resources are written in Python. The machine learning resources available in Python allow the development of algorithms as a high-level *computational graph*, which represents a directed graph. The nodes of a computational graph can be computation inputs, constants, or learned variables. The edges of the graph represent operations such as "add" or "batch-wise multiplicaton" on nodes. Using a market simulator implemented in Java and a DRL algorithm in Python created a need to develop an agent which can interact in multiple computing languages. In collaboration with another University of Michigan student, I built an interface which uses a language-agnostic agent implemented Tensorflow (Abadi et al., 2015). This agent uses DRL to train in Python, and generates experiences in Market-Sim.

Figure 3.1 depicts how the agent trains in Python and trades in Market-Sim. The agent uses a DRL algorithm to train its computational graph in Python, then it passes its graph and corresponding parameters to Market-Sim. The graph and parameters are loaded into Market-Sim and the agent generates and logs experiences. These logged experiences are passed back to the training process after the trading period ends. The process repeats itself and can use the new logged experiences to train.

3.4 Empirical Game-Theoretic Analysis

The outcome of a simulated market model varies depending on the traders' strategy selection and parameter settings. An agent's profit by using one trading strategy can

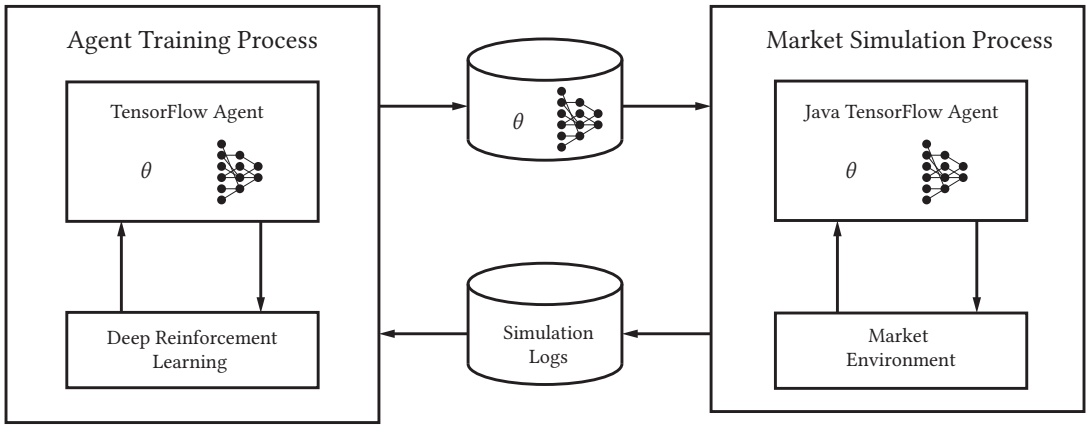


Figure 3.1: Overview of implementing a DRL trading agent in Market-Sim. On the left, the training process to periodically improve the agent's trading strategy. The agent exports its computational graph and parameters to Market-Sim. The agent participates in Market-Sim and logs its experiences. The agent returns those logs to the training process.

drastically change given which strategies the other agents in the market are using. Therefore, it is beneficial to determine the optimal strategy for a trading agent given the strategies the other agents use. To analyze a market model in this way, I model each market as a game. In this the market game, the traders represent players, where each player adopts a strategy. A *strategy* in this market game is a trading strategy. An agent's *payoff* is the final valuation of the agent's holdings plus net cash flow at the end of the game. Each player's final payoff depends on the actions the other players in the game, because if, when, and what price a player's order transacts at relies on the orders of other players.

In each game players are split into *roles*, such as background traders and a manipulator, where each role can choose trading strategies from a specified set. The games I analyze are *role-symmetric games*, where players are split into roles with a specified strategy set. Players within a role can select a *pure strategy*, which is when a player only plays on strategy. Players within a role can also select a *mixed strategy*, which is when a player draws a strategy to play from a distribution over the strategy set. A *strategy profile* is a strategy assignment for each role in the game. A *pure strategy profile* is when every role plays a pure strategy every simulation run. A *mixed strategy profile* happens when at least one role plays a mixed strategy. I find the *Nash equilibrium* for some market games, which occurs when no player can increase its final

payoff by deviating to another strategy.

Empirical game-theoretic analysis (EGTA) is a method to analyze games (Wellman, 2016). A market simulator characterizes the games I analyze with EGTA. EGTA uses a significant amount of simulation data to estimate the market game model. Market-Sim and ABIDES are used to collect simulation data to approximate different market games. I use EGTAOnline, a tool developed by Cassell and Wellman (2012), to manage the process of collecting simulated data and game analysis. First, the game is restricted to a subgame with only pure strategy profiles. A *subgame* is a game where a subset of strategies is used for each role. EGTA estimates payoffs for each role as the sample average of payoffs from a large number of simulation runs. EGTA identifies *candidate* equilibria in a subgame, then incrementally adds strategies back to each roles strategy set. As the subgame grows, EGTA searches for another strategy profile that would be a beneficial deviation from a candidate for at least one role. A candidate is *refuted* if EGTA finds a beneficial deviation. A candidate is *confirmed* if EGTA cannot find a beneficial deviation. When a candidate is confirmed by EGTA has reached quiescence and the candidate is determined to be a Nash equilibrium for the game.

3.4.1 Game Reduction

The number of profiles grows exponentially as the number of players and strategies increase. This makes it infeasible to perform the necessary simulations to successfully estimate such a large game model. To address this issues, I use *deviation-preserving reduction* (DPR), which aggregates players within a role to approximate a many-player game (Wiedenbeck and Wellman, 2012). DPR defines the payoffs in the reduced game over a strategy profile by the payoffs from the original game. I select number of players in the reduced and original games to ensure scalability between the two games.

3.4.2 Deviating from Equilibrated Games

I implement an agent which generates its trading strategies through deep reinforcement learning (DRL) in Chapter V and VI. An agent that learns strategies through DRL estimates the best strategy to play by training a policy that selects which action to take given the current market environment. In a market game an action is submitting an order. Estimating a game model with EGTA and training a DRL algorithm's policy for trading both require collecting large amounts of simulation data

to approximate the payoffs of a strategy. Training a DRL algorithm attempts to find the most profitable trading strategy while other agents have predetermined strategies. EGTAOnline estimates the Nash equilibrium of a market game where players play heuristic strategies. EGTAOnline cannot handle finding Nash equilibrium when some players' strategies are derived with DRL. Therefore, I define an alternative method in a game where a player in the game uses DRL to learn strategies. I integrate strategies learned through DRL into the process of learning the game model to analyze strategy profiles with the following process. I first equilibrate a market game with players who only use heuristic strategies. Then, I replace one agent in the game with a DRL agent, and train that agent until it learns a profitable trading strategy. Once the DRL agent has learned a trading strategy in the original equilibrated game, it is possible that the strategy profile selected by the other agents is no longer the Nash Equilibrium. Therefore, I test single-player strategy deviations of the background traders while the DRL agent continues to play its learned strategy. If there is a beneficial single-player deviation s_2, then I test a variety of mixed strategies containing s_1 and s_2. For each mixed strategy I test a different probability of a player playing s_1 versus s_2. If the original equilibrate strategy is a pure strategy s_1 then this mixed strategy exploration may include when the probability of a background agent playing each strategy to be $P(s_1) = 0.5$ and $P(s_2) = 0.5$. If the background traders deviated to another strategy profile, I retrain the DRL agent to learn a new trading strategy. I then repeat this deviation process with another single-player deviation. If the original equilibrium strategy was a pure strategy, the background agents now explore deviating to mixed strategies distributed over three strategies. If the background traders again deviated to another strategy profile, I again retrain the DRL agent to learn a new trading strategy.

CHAPTER IV

Stability Effects of Arbitrage in Exchange Traded Funds

An index-based exchange traded fund (ETF) with underlying securities that trade on the same market creates potential opportunities for arbitrage between price deviations in the ETF and the corresponding index. I examine whether ETF arbitrage transmits small volatility events, termed mini flash crashes, from one of its underlying symbols to another. I address this question in an agent-based, simulated market where agents can trade an ETF and its two underlying symbols. I explore multiple market configurations with active and inactive ETF arbitrageurs. Through empirical game-theoretic analysis, I find that when arbitrageurs actively trade, background traders' surplus increases because of the increased liquidity. Arbitrage helps the ETF more accurately track the index. I also observe that when one symbol experiences a mini flash crash, arbitrage transmits a price change in the opposite direction to the other symbol. The size of the mini flash crash depends more on the market configuration than the arbitrageurs, but the recovery of the mini flash crash is faster when arbitrageurs are present.

4.1 Introduction

An *exchange traded fund* (ETF) is a portfolio of securities that trades on the stock market. The *underlying securities* in an ETF's portfolio can be any traded entity, such as stocks, bonds, or commodities. ETFs have become a popular investment vehicle, as they provide convenient access to portfolio trades, offering average investors apparent liquidity (Antoniewicz and Heinrichs, 2014). Investing in an ETF, rather than each

security in its portfolio, requires fewer trades, and therefore offers the opportunity for diversification with lower trading costs (Golub et al., 2013).

I focus on *index-based ETFs*, which are ETFs whose underlying securities are based on a *market index*. Some examples of market indexes are the Standard and Poor's (S&P) 500 and Dow Jones Industrial Average (DJIA) which track market performance, or the Volatility Index (VIX) which tracks market volatility. Though designed to track the index, the actual price of the index-based ETF is determined through trading it as a symbol on the stock market. When the market is open, participants can simultaneously observe the trading price of the ETF and calculate the index. Any disparity composes a potential arbitrage opportunity between the ETF and its underlying securities.

Arbitrage trading can help an ETF's price track its corresponding market index (Antoniewicz and Heinrichs, 2014). It may also introduce or reinforce other dependencies. For example, arbitrage may tether the price volatility of the ETF's underlying symbols (Ben-David et al., 2015). The rise in market-wide price volatility in recent years has raised concerns about economic growth and investor trust (Carlson, 2019; Li, 2019). Price co-movements in ETF portfolios have led some to question the role ETFs play in amplifying volatility (Ben-David et al., 2015; Pagano et al., 2019). I study this issue, focusing specifically on *mini flash crashes*, which are short volatility events where the price rapidly changes, then quickly reverts (Shearer, 2020). ETF arbitrage may channel mini flash crashes from one underlying symbol to others. ETF arbitrage may also have other effects, for example on the distribution of surplus among traders.

To explore these questions, I develop an agent-based model (ABM), populated by trading agents implementing algorithms commonly employed in agent-based finance literature. My market consists of multiple stocks which compose the portfolio of an ETF. I study my simulated market with active and inactive ETF arbitrageurs. I then induce a mini flash crash to one of the underlying securities, by injecting a trader who submits a series of large marketable limit orders (a common pattern of behavior triggering such mini flash crashes (Braun et al., 2018; Golub et al., 2017; Shearer, 2020)). I then analyze the effect of arbitrage activity on the market response.

Using agent-based simulation allows me to examine otherwise identical market environments with and without the presence of ETF arbitrage. I explore the impact of mini flash crashes by comparing the two scenarios, with strategic equilibration of trading agents under each setting. I implement two strategies that trade on arbitrage opportunities between the ETF and its underlying symbols. This allows me to observe

whether trading strategies dependent on an ETF and its portfolio directly impact their underlying symbols' price volatility.

I employ a simulated market model using a standard limit order book and message system similar to the US stock exchange NASDAQ (Byrd et al., 2020). My market contains two symbols which compose an ETF's portfolio. The ETF also trades on the market. My model is populated with 27 background trading agents and four arbitrageurs. A conservative and an aggressive arbitrageur trade only on the ETF, and a conservative and an aggressive arbitrageur trade on both ETF and underlying symbols. I also use one impact agent to submit a series of large, marketable sell orders to create a mini flash crash.

Through empirical game-theoretic analysis, I determine the optimal trading strategies for all agents when arbitrageurs are active and inactive, in order to examine the impact of ETF arbitrage on market welfare and volatility. Arbitrageurs are highly profitable and background agents' average surplus is impacted significantly by the arbitrage activity. Background traders' final payoffs increase with active arbitrage because arbitrageurs increase liquidity by submitting marketable orders. Active arbitrage also increases the volatility of its underlying symbols around events like mini flash crashes. When one underlying symbol experiences a mini flash crash, the other symbol experiences a price change in the opposite direction when arbitrageurs are active. With active arbitrage there is a faster price reversion of the mini flash crash in the symbol which originally experienced the event. The competitiveness of the background traders, rather than the arbitrageurs, influences the magnitude of the mini flash crash. The price of the index, ETF, and symbol that experiences the mini flash crash are lower than the price preceding the event. When arbitrageurs are active, the average price of the other underlying symbol is higher than the price preceding the event. Overall, the study demonstrates the effect of ETF arbitrage on welfare and how ETFs can spread volatility events through their portfolios and contribute to market-wide volatility.

This chapter is organized as follows. I discuss prior work in Section 4.2. In Section 4.3 I provide an overview of an ETF's market structure and the arbitrage opportunities this structure creates. I then describe the market mechanism for my simulated ETF environment in Section 4.4. Section 4.5 presents my findings on the impacts of mini flash crashes in a market with active and inactive ETF arbitrageurs. I conclude in Section 4.6.

4.2 Related Work

Most prior work uses historical data and quantitative models to analyze the impact of ETFs on the volatility of their underlying symbols. Using high frequency data, Anatolyev et al. (2020) found that ETF arbitrage distorts reactions to market shocks in the US stock market. Ben-David et al. (2015) constructed a quantitative model and used historical data to conclude that ETFs increase the volatility of their underlying symbols. Da and Shive (2017) also analyzed historical data and found that ETFs contribute to price co-movement in the symbols in their portfolios. In contrast, Madhavan and Morillo (2018) found that underlying symbol price co-movements correlate with macro-market movements, rather than the presence of the ETF. Lastly, Lynch et al. (2019) found that an implementable trading strategy could generate profits using ETF arbitrage given realistic portfolio price co-movements.

There are also previous studies using historical data to examine what leads to ETF arbitrage opportunities. Box et al. (2021) found that a price shock or order imbalance in an ETF's underlying portfolio typically precedes ETF arbitrage opportunities. Marshall et al. (2013) found that spreads increase before ETF arbitrage opportunities.

My agent-based market simulation builds on that of Byrd et al. (2020). Simulated market models allow other traders to strategically react given the current state of the market, which changes each iteration. It also enables me to control factors of influence, and study the effect of single factors (e.g., presence of ETF arbitrage).

Numerous other ABM studies have examined order and trade activity around volatility events. Several have replicated aspects of the Flash Crash on May, 6 2010 to determine which trading practices contributed to the price drop and recovery (Paddrik et al., 2012; Veryzhenko and Oriol, 2012; Vuorenmaa and Wang, 2014). Some other ABM studies (Bayraktar and Munk, 2017; Leal and Napoletano, 2019; LeBaron et al., 1999) have explored abstract mini flash crashes and price bubbles to provide insight into what might exacerbate these events. Paulin et al. (2018) incorporated a dependency network to study the micro and macro impacts of flash crashes across multiple securities.

Two other prior ABM studies analyze ETF arbitrage. Mizuta (2019) examined the impact of trading costs and ETF arbitrage, finding that lower trading costs and higher price volatility lead to more arbitrage opportunities, more trading volume, and stronger correlation between the index and ETF price. Torii et al. (2016) model an index fund and two underlying symbols, and examine the impacts of ETF arbitrage

on response to a downward volatility event in one of the symbols.

The model of Torii et al. (2016) is in fact quite similar to ours, but with a few key differences. First their background traders employ technical trading strategies (i.e., consider price trends), whereas my agents are purely fundamental traders. Second, they generate volatility events by directly manipulating the fundamental value. I initiate a different type of volatility event, a mini flash crash, through an agent who quickly submits multiple large orders. In a mini flash crash, the price momentarily drifts from the asset's true value because of trading activity. Such events commonly occur in the real stock market when a trader who wants to leave a large position and starts to sell off all of their holdings (Braun et al., 2018; Golub et al., 2017; Shearer, 2020). These two methods of creating volatility events exhibit entirely different order-book dynamics, generating qualitatively different arbitrage opportunities and outputs, and differences in response after the shock. Third, the prior work does not model the primary market of an ETF, which is also an essential element in defining the performance of arbitrageurs. Despite these differences, both the study of Torii et al. (2016) and my own find that when one underlying symbol experiences a downward price shock, the other underlying symbol experiences a price shock in the opposite direction.

4.3 ETF Market Structure

The market structure of an ETF is unique in that ETFs actually trade on two markets. An ETF trades on the stock market like any other stock, but rather than its value being derived from a company, its value is based on a portfolio of other securities. Similar to other portfolio management funds, such as hedge funds, an ETF also allows investors to accrue and liquidate fund holdings. This fund market is referred to as an ETF's primary market, and the stock market is the ETF's secondary market. Since an ETF trades on two different markets, it has two potentially deviating prices. This creates an arbitrage opportunity for market participants able to detect and act on deviations between the ETF's two prices.

4.3.1 Primary Market

An ETF is a portfolio management fund, and like various types of these funds, only a select few participants, commonly referred to as authorized participants (AP), can invest and divest in an ETF at a time and price determined by its fund managers

(Novick et al., 2017). An AP can trade daily at the *Net Asset Value* (NAV), which is the price of the ETF in the primary market. For an index-based ETF, the NAV is calculated by finding the true value of the market index at the close of the stock market. Once the NAV is determined, the fund will accept orders until a predetermined time when all orders are executed. These orders are referred to as *basket orders*, which are composed of shares of the ETF or shares of the underlying symbols. APs submit basket orders to the fund manager in return for shares of underlying symbols or shares of the ETF, respectively (Gastineau, 2004). To submit a basket order, an AP must first acquire shares of the ETF or underlying symbols on the stock market.

A notable aspect of the ETF primary market is that like other portfolio management funds there are high barriers to entry. APs tend to be large institutional investors because they need to have a lot of capital and resources to create basket orders on an ETF's primary market (Antoniewicz and Heinrichs, 2015). While the primary market is fairly exclusive, ETFs are unique for portfolio management funds in that their secondary markets are much more accessible to a variety of investors.

4.3.2 Secondary Market

In contrast to other portfolio management funds, ETFs also trade on a secondary market, which is the stock market. A stock share of an ETF corresponds to a small portion of the ETF's portfolio. On the stock market, an ETF trades like any other symbol where traders can submit orders to buy or sell shares of the ETF on a stock exchange (Engle and Sarkar, 2006). When a trader submits an order, it will either transact with an order on the opposite side, or will rest in a limit order book until it can be transacted or the trader cancels the order. Since an ETF has an active order book throughout a trading day, it also has a trading price on its secondary market. I define two trading prices of an ETF, the *bid trading price* and *ask trading price*, as the highest bid price and lowest offer price, respectively, for visible volume in its secondary market's limit order book.

An ETF's secondary market gives an ETF much lower barriers to entry than other portfolio management funds (Poterba and Shoven, 2002). The accessibility of the secondary market is a principal reason for the popularity of ETFs (Golub et al., 2013). Participants in the primary market can also trade in the secondary market.

4.3.3 Arbitrage between the Primary and Secondary Markets

An ETF trades at the NAV and trading price on its primary and secondary markets, respectively. Even though the NAV is only calculated once per day, it can be estimated throughout the secondary market's trading day by calculating the market index. When the bid trading price rises above the market index, a trader can make a profit by buying the underlying symbols and selling the ETF. The reverse is true as well, so when the ask trading price falls below the market index, a trader can make a profit by buying the ETF and selling the underlying symbols. Arbitrage between these two prices is encouraged because it helps force the trading price to more closely track the index (Antoniewicz and Heinrichs, 2014).

While any secondary market participant can arbitrage price deviations between the market index and trading price, this is a particularly beneficial arbitrage opportunity for participants with access to both the primary and secondary market (Poterba and Shoven, 2002). Such a participant can arbitrage in the secondary market, then submit basket orders in the asset it is long in for shares of the asset it shorted.

4.4 Market Mechanism

4.4.1 ABIDES

To investigate the potential for an ETF to spread a mini flash crash from one symbol to another, I employ ABIDES, a market model discussed in Chapter III. The platform provides a continuous double auction market with securities priced in cents, a set of typical background agents, and a kernel which drives the simulation with nanosecond resolution while permitting sparse activity patterns to be efficiently computed.

For the ETF secondary market, I use the provided ABIDES exchange agent, which operates in a manner similar to the NASDAQ. The market is open from 09:30 to 16:00, lists any number of securities for trade, and provides a distinct order book mechanism for each security. The exchange accepts limit orders of any share volume, and cancellation of same, and transacts (including partial execution) those orders against a security's limit order book with a typical price-then-FIFO matching algorithm. The exchange responds to requests for market hours, last trade prices, and market depth quote requests, with depth one representing the current best bid and ask.

4.4.2 Market Index

I define this model's *bid market index* at time t as the sum of the highest available bid price of a bundle of stocks:

$$\iota_{t,b} = \sum_{i=1}^{n} b_{i,t}. \tag{4.1}$$

I also define this model's *ask market index* at time t as the sum of the lowest available ask price of a bundle of stocks:

$$\iota_{t,a} = \sum_{i=1}^{n} a_{i,t}. \tag{4.2}$$

In this work traders only submit orders of one size, so the bid and market index are not weighted by size and only consider the price of a single unit of stock. I also only examine baskets consisting of two stocks, so $n = 2$. I based my market index off the DJIA, which is a US stock market index that tracks 30 large market capitalization stocks. However, my index is simpler than the DJIA, which sums the prices of these 30 stocks and divides by the DJIA divisor. My market index calculation simplifies the basket orders in the ETF primary market, because one share of the ETF is equivalent to a basket of one share of each symbol in its portfolio.

4.4.3 Primary Market in ABIDES

After the exchange agent stops accepting orders, the primary market receives the close price, p_i of each underlying symbol, s_i in its ETF's portfolio, where the *close price* is the price of a symbol's last trade on the secondary market. The primary market uses these closing prices p_i to calculate the value of the index and uses this value as the NAV. In my market, the NAV is calculated by:

$$NAV = \sum_{i=1}^{n} p_i. \tag{4.3}$$

Then the primary market opens for basket orders. Every basket order it receives is executed at the NAV, and the agent is notified of its transaction.

4.4.4 ETF Fundamental Calculation

Most symbols in my market model have independent fundamental values defined in Section 3.1.2, but an ETF's fundamental is dependant on the value of other symbols in the market. My ETF is an index-based ETF, and I use the index from Equations (4.1) and (4.2). Using this index and the value of the fundamental $\hat{r}_{i,t}$ of symbol s_i, I can find the fundamental of the ETF at time t by:

$$\hat{r}_t = \sum_{i=1}^{n} \hat{r}_{i,t}.$$

4.4.5 ETF Arbitrage Agent

I develop four strategies for arbitrage between the ETF and its underlying symbols. Like the background agents, arbitrage agents enter the market at Poisson arrival rate λ_a. On arrival at time t, a conservative agent calculates the difference between the best ETF bid price, $b_t^{(\text{ETF})}$ and the ask market index $\iota_{t,a}$:

$$\Delta_{t,1} = b_t^{(\text{ETF})} - \iota_{t,a}.$$

A conservative arbitrage agent must also find the difference between the bid market index, $\iota_{t,b}$, and the best ETF ask price, $a_t^{(\text{ETF})}$:

$$\Delta_{t,3} = \iota_{t,b} - a_t^{(\text{ETF})}.$$

A more aggressive agent finds the difference between the ETF mid price, $m_t^{(\text{ETF})}$ and the mid market index $\iota_{t,m}$:

$$\begin{aligned} \Delta_{t,2} &= m_t^{(\text{ETF})} - \iota_{t,m}. \\ \Delta_{t,4} &= -\Delta_{t,2}. \end{aligned}$$

All arbitrageurs submit only marketable orders, so their sales are at the bid and their buys at the ask. I denote an arbitrageur that uses the conditions $\Delta_{t,2}$ and $\Delta_{t,4}$ as aggressive because they use the midpoint to determine whether to trade or not. The agents in the this market cannot trade at the midpoint, but the midpoint is arguably the best metric to predict NAV and ETF close price. Prior work also uses the midpoint when studying ETF arbitrage (Box et al., 2021). Therefore, I consider both sets of conditions to study types of arbitrage and which are more profitable

during volatility events. The arbitrageur that uses conditions $\Delta_{t,2}$ and $\Delta_{t,4}$ are denoted as aggressive because this set of conditions are more likely to be met than $\Delta_{t,1}$ and $\Delta_{t,3}$.

ETF Single Asset Arbitrageur This single asset arbitrageur trades exclusively on the ETF security. It decides when to trade based on a threshold $\varepsilon \geq 0$:

$$\begin{cases} \Delta_{t,1} > \varepsilon & \text{Sell ETF,} \\ \Delta_{t,3} > \varepsilon & \text{Buy ETF.} \end{cases}$$

An alternative, less conservative version of this arbitrageur makes decisions based on the midpoint prices ($\Delta_{t,2}$ and $\Delta_{t,4}$) instead.

ETF Multiple Asset Arbitrageur The multiple asset arbitrageur trades both the ETF and its underlying symbols. Its trades are also triggered by a threshold $\varepsilon \geq 0$:

$$\begin{cases} \Delta_{t,1} > \varepsilon & \text{Sell ETF and buy underlying symbols,} \\ \Delta_{t,3} > \varepsilon & \text{Buy ETF and sell underlying symbols.} \end{cases}$$

An alternative, less conservative version of this arbitrageur makes decisions based on the midpoint prices ($\Delta_{t,2}$ and $\Delta_{t,4}$) instead.

These agents also trade on the primary market. When the primary market opens, they receive the NAV from Equation 4.3, then decide if they should submit basket orders by comparing the NAV and close price of the ETF, $p^{(\text{ETF})}$. More formally:

$$\begin{cases} NAV - p^{(\text{ETF})} > 0 & \text{ETF shares } \rightarrow \text{ underlying symbol shares,} \\ p^{(\text{ETF})} - NAV > 0 & \text{Underlying symbol shares } \rightarrow \text{ ETF shares.} \end{cases}$$

If they submit basket orders, then they can hopefully end the day net zero. The market makers need to be the fastest agents in the system in order to be profitable when trading on so many symbols.

4.5 Results

4.5.1 Market Environment Settings

I examine a variety of market environments to analyze the robustness of my results. Each environment contains one exchange agent, one ETF primary market agent, two symbols, and an ETF whose portfolio is composed of the two other symbols. The exchange agent accepts orders between 12:30 and 13:30. The ETF primary market accepts orders between 17:00 and 17:01. Following Wang et al. (2018), I consider three market environments vary by market shock σ_s^2 and observation noise σ_n^2. The first consists of low shock and high observation noise (**LSHN**) with $\sigma_s^2 = 5 \times 10^4$ and $\sigma_n^2 = 10^7$. The second consists of medium shock and medium observation noise (**MSMN**) with $\sigma_s^2 = 5 \times 10^5$ and $\sigma_n^2 = 5 \times 10^6$. Lastly, the third holds high shock and low observation noise (**HSLN**) with $\sigma_s^2 = 5 \times 10^6$ and $\sigma_n^2 = 10^6$.

For each non-ETF symbol, I generate a fundamental (3.5), with mean $\mu = 10^5$, reversion $\gamma = 1.67 \times 10^{-13}$. The sparse fundamental experiences a series of megashocks throughout the trading period, and these arrive according to a Poisson distribution with $\lambda = 2.78 \times 10^{-13}$. I draw the size of these megashocks from a binomial normal distribution with means $\mu_{s,1} = 0$ and $\mu_{s,2} = 10^3$, and varying values of σ_M^2.

This market is populated with 27 background agents, where each agent is randomly assigned one symbol with equal probability to trade for the duration of each market run. Table 4.1 specifies the strategies of the background agents. The background agents arrive to the market according to a Poisson distribution with $\lambda_a = 10^{-11}$. These agents submit orders of size $q = 100$, but can hold a maximum number of units at any time $q_{\max} = 10^3$. When the background agents consider past transactions to update their estimate of the fundamental, they use a variance of $\sigma_p^2 = 10^3$. Lastly, the private value variance is $\sigma_{PV}^2 = 5 \times 10^6$.

I create a mini flash crash in one underlying symbol with a single impact agent.

Table 4.1: Strategies employed by the background traders, where each agent chooses their desired surplus between $R_{\min}$ and $R_{\max}$, and have hyperparameter η.

Strategy	$\mathbf{ZI_1}$	$\mathbf{ZI_2}$	$\mathbf{ZI_3}$	$\mathbf{ZI_4}$	$\mathbf{ZI_5}$	$\mathbf{ZI_6}$
$R_{\min}$	2000	2000	2500	3000	3000	3500
$R_{\max}$	2500	3000	3000	3500	4000	4000
η	1.0	0.8	1.0	1.0	0.8	1.0

This impact agent is assigned an underlying symbol to trade on with equal probability at the beginning of each market run. It then submits 5 trades beginning at 13:00 with size 100 and 6 seconds between each trade.

I implement four ETF arbitrage strategies, with one conservative ETF one asset (SA-C) agent, one aggressive ETF single asset (SA-A) agent, one conservative ETF multiple asset (MA-C) agent, and one aggressive ETF multiple asset (MA-A) agent. When arbitrageurs actively trades, all arbitrageurs use a strategy with $\varepsilon = 10^3$. However, when the arbitrageurs are inactive, they only exercise a strategy where $\varepsilon = 10^{12}$. Both arbitrageurs submit orders of size $q = 100$. All ETF arbitrageurs arrive to the market according to a Poisson process with $\lambda_a = 5 \times 10^{-3}$.

4.5.2 EGTA Process

I model my market as a role-symmetric game, where players are divided into five roles: background traders, SA-C, SA-A, MA-C, and MA-A. I utilize *deviation-preserving reduction* (DPR), which approximates a many-player game by aggregating a game with fewer players (Wiedenbeck and Wellman, 2012), because a game grows exponentially in players and strategies. DPR has shown to generate good approximations of the full game in multiple settings.

My game consists of 27 background traders, one of each arbitrage role, and reduces to three background traders and one of each arbitrageur when using DPR. These quantities of traders ensure that the required aggregations from DPR come out as integers. In this setting one background agent deviates to a new strategy while the other 26 background agents are further reduced to two. For a specific strategy profile, I sample between 200 and 10,000 simulation runs to reduce sampling error from stochastic market features.Appendix A presents the details of strategies employed in the Nash Equilibria.

4.5.3 Impact on Market Welfare

I use EGTA to analyze the impact of ETF arbitrage on market welfare across varying market environments. Figure 4.1 depicts the surplus of each role when ETF arbitrageurs are active, when arbitrageurs are active but background agents have not adjusted to their presence, and when arbitrageurs are inactive in each market setting. The background agents are unadjusted to active arbitrageurs when they utilize the strategies optimal with inactive arbitrageurs in a setting when those agents

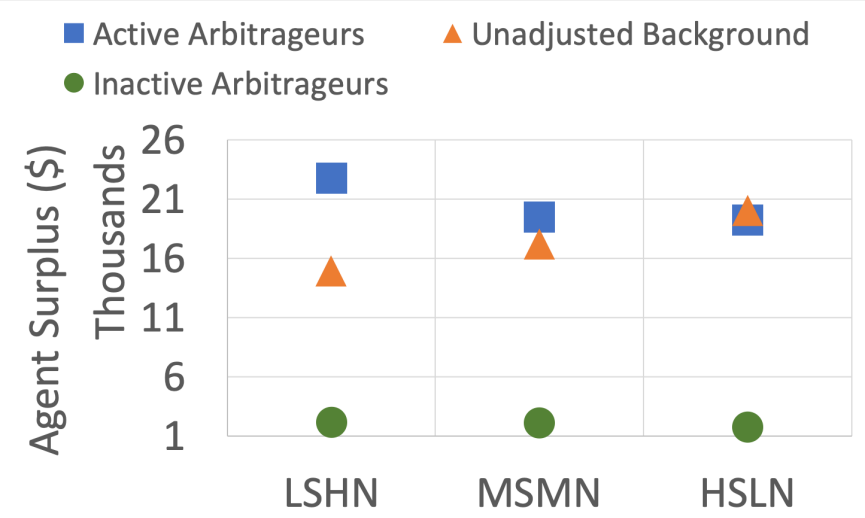


Figure 4.1: The average payoff over 200 simulation runs for the background agents when arbitrageurs are active and inactive in the three market environments.

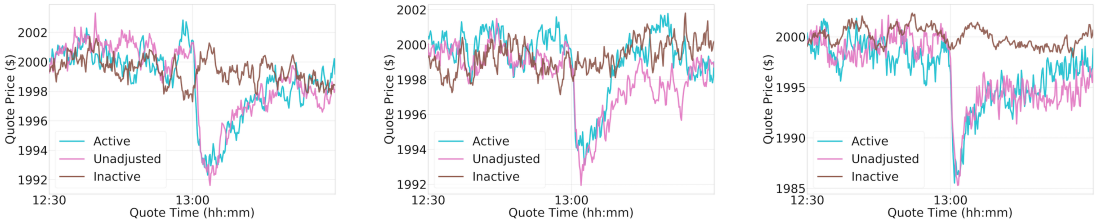
are actually active. An *inactive arbitrageur* is an arbitrageur that does not submit marketable orders during the trading period.

Background agents are better off when arbitrageurs are active. When arbitrageurs are active they provide more marketable liquidity, particularly during a mini flash crash where there are many arbitrage opportunities. Thus, background traders have more opportunities to trade when arbitrageurs are active. These traders submit less competitive orders when arbitrageurs are active. A less competitive ZI strategy is when the agent selects a higher desired surplus range. When arbitrageurs are active, they are willing to trade with these high-margin orders, so the background traders' profit increases. The background traders realize similar payoffs when they do not adjust to arbitrage.

When the arbitrageurs are inactive, the surplus of the SA-Cs, SA-As, MA-Cs, and MA-As is always zero. ETF-MAs have a higher surplus than ETF-SAs because they have the opportunity to trade on all symbols, and the ETF primary market where ETF-MAs receive the difference in value between the ETF and index.

4.5.4 Impact on Symbol Price

I analyze ETF arbitrageurs' impact on market volatility, when one underlying symbol experiences a mini flash crash. To assess this impact I examine and compare the price of the underlying symbols in environments with active arbitrageurs, active



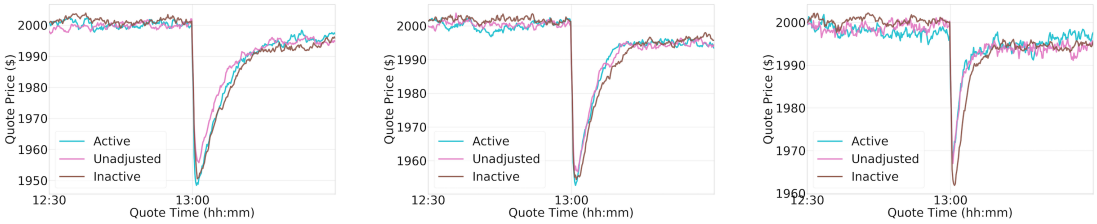
(a) LSHN, Active Arbitrage: ETF (b) MSMN, Active Arbitrage: ETF (c) HSLN, Active Arbitrage: ETF

Figure 4.2: Average price time series over 200 simulation runs of the ETF. The ETF in a market is meant to track the market index. The ETF is only present in three of the market environments.

arbitrageurs and unadjusted background agents, and inactive arbitrageurs. An environment with active arbitrageurs and unadjusted background agents happens when background agents do not recalibrate their strategies to consider arbitrage. In each figure I represent the price as the midpoint price over 200 simulation runs.

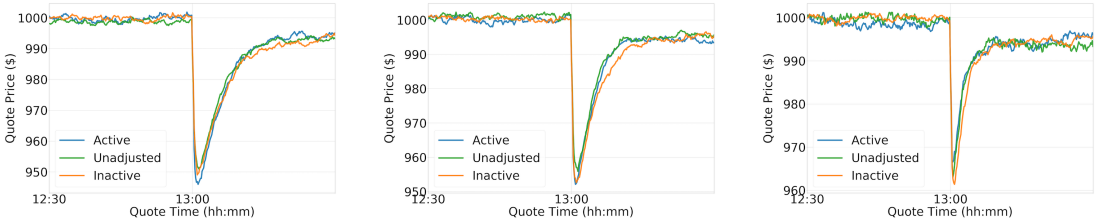
The price of the ETF should track the market index. Figure 4.2 shows the midpoint price of the ETF and Figure 4.3 depicts the market index of each environment. This allows me to see how arbitrage impacts the ETF trading price and market index, and what trading opportunities arbitrageurs have around the mini flash crash in an underlying symbol. The ETF trading price sees a decrease in price when arbitrageurs are active. The market index crashes a large amount in every environment at the time of the mini flash crash. In environments active and inactive arbitrage, the index drops because as the midpoint price of an underlying symbol drops, the market index or the sum of prices s_0 and s_1 also drops. When ETF arbitrageurs are present, as the market index falls below the ETF price, the arbitrageurs sell the ETF and buy the underlying symbols. As the arbitrageurs sell the ETF, the bid side of the order book reduces and the price drops. When the arbitrageurs buy the underlying symbols they absorb the ask side of the order book, which leads to the market index recovering quicker and back to a higher price when an ETF is present. Thus, arbitrage helps the ETF track the index.

In all market environments, the impact agent causes a distinct mini flash crash in the underlying symbol. This symbol experiences similar trends to the market index, where more competitive background agents creates a smaller spread and smaller price drop. Figure 4.4 shows each environment in a reduced time frame around the mini flash crash, and active arbitrageurs, inactive arbitrageurs, and unadjusted background



(a) LSHN, Active Arbitrage: Index (b) MSMN, Active Arbitrage: Index (c) HSLN, Active Arbitrage: Index

Figure 4.3: Average time series over 200 simulation runs of the market index. This index is composed of two symbols. An underlying symbol experiences a mini flash crash at 13:00.

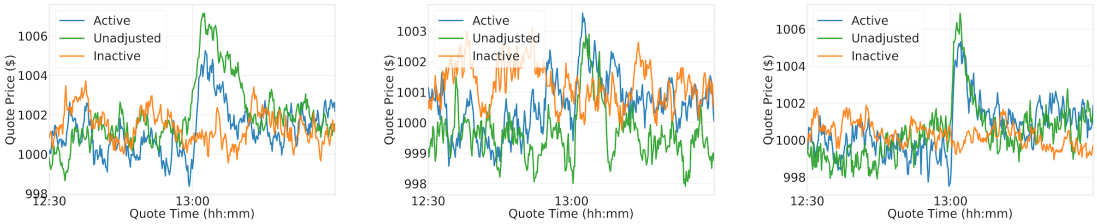


(a) LSHN: s_0 (b) MSMN: s_0 (c) HSLN: s_0

Figure 4.4: Average price time series over 200 simulation runs of underlying symbol, s_0, that is randomly selected to experience a mini flash crash at 13:00 through an impact agent submitting a series of large trades. This shows active arbitrageurs, inactive arbitrageurs, and unadjusted background traders on the same plot.

agents are depicted together. The mini flash crash recovers faster with active arbitrage because the arbitrageurs submit marketable buy orders to the underlying symbol and marketable sell orders to ETF, causing the price of the index and underlying symbols to rise faster. In each environment, the price of this symbol does not recover to the price before the mini flash crash, though the price does revert to a higher level when arbitrageurs are active.

Figure 4.5 depicts the average price of the underlying symbol where the impact agent does not trade. When background traders are more competitive, there is a small but distinct upward price movement at the time of the mini flash crash in the other symbol. This happens because the arbitrageurs buy the ETF's underlying symbols when the index is lower than the ETF trading price, causing the price increase. In environments with arbitrage, the average price is higher than the price preceding the mini flash crash. The increase in price This implies that the presence of ETF



(a) LSHN, Active Arbitrage: s_1 (b) MSMN, Active Arbitrage: s_1 (c) HSLN, Active Arbitrage: s_1

Figure 4.5: Average price time series over 200 simulation runs of underlying symbol without a mini flash crash. An impact agent does not trade on this symbol.

arbitrageurs can impact the trading price of a symbol because of trading activity independent of the symbol itself.

4.6 Conclusion

I analyze an agent-based, simulated market model with a stock market and ETF primary market. I explore varying market environments that contain an ETF and two symbols which compose the ETF's portfolio. The market is also populated with numerous background agents, an impact agent which creates a mini flash crash, and ETF arbitrage agents. To determine the impact of ETF arbitrage, I examine equilibria in each environment under settings when arbitrageurs are active and inactive. The arbitrageurs are extremely profitable when they actively trade, and the surplus of the background agents increases when they are conservative and decreases when they are more competitive. I also find that this type of arbitrage may transmit a volatility event, like a mini flash crash, throughout its underlying symbols. The other underlying symbol experiences a price change in the opposite direction of the other underlying symbol at the time of the mini flash crash. The magnitude of the mini flash crash in the original symbol is impacted more by the competitiveness of the background traders than the arbitrageurs.

Most real-world ETFs represent portfolios with many symbols. My study models an ETF with only two symbols, in order to focus the effect on a single relationship. I believe this captures qualitative properties of realistic ETFs as well, but it could be that further insights would be revealed by extending the model to cover more symbols. This could be readily incorporated in my ABM, with linear impact on computational cost of simulation, though perhaps imposing somewhat more complexity on the

arbitrage strategy and the analysis of results. An ETF with a larger portfolio could cause arbitrage to have a larger impact on the ETF than the underlying symbols, so arbitrage may help the ETF track the index, but introduce less volatility to the underlying symbols.

Another potential limitation is in the space of background trading strategies. I focus exclusively on background agents that use a ZI strategy which consider previous transactions in their estimate of an asset's fundamental value. It could be beneficial to utilize more trading strategies dependent on the order book and price movement, such as market makers or trend followers, because these agents might exacerbate mini flash crashes.

This work provides insight into the impact of ETFs on market welfare, market volatility, and stock valuation. Previous studies have used historical data to examine associations between ETF activity and price volatility in its underlying symbols. With agent-based simulation I are able to examine causality through a direct A/B test in market environments with and without active ETF arbitrage. I find other agents are better off with arbitrage if they are more conservative, but arbitrage reduces their profits if they trade more competitively. An implication is that inclusion in an index ETF may impact the pricing of a stock without any actual change in the stock's fundamental value, and solely due to trading activity independent of the stock itself.

CHAPTER V

Benchmark Manipulation

Financial benchmarks estimate market values or reference rates used in a wide variety of contexts, but are often calculated from data generated by parties who have incentives to manipulate these benchmarks. Since the London Interbank Offered Rate (LIBOR) scandal in 2011, market participants, scholars, and regulators have scrutinized financial benchmarks and the ability of traders to manipulate them. I study the impact on market welfare of manipulating transaction-based benchmarks in a simulated market environment. My market consists of a single benchmark manipulator with external holdings dependent on the benchmark, and numerous background traders unaffected by the benchmark. I explore two types of manipulative trading strategies: zero-intelligence strategies and strategies generated by deep reinforcement learning. Background traders use zero-intelligence trading strategies. I find that the total surplus of all market participants who are trading increases with manipulation. However, the aggregated market surplus decreases for all trading agents, and the market surplus of the manipulator decreases, so the manipulator's surplus from the benchmark significantly increases. This entails under natural assumptions that the market and any third parties invested in the opposite side of the benchmark from the manipulator are negatively impacted by this manipulation.

5.1 Introduction

Financial benchmarks play a pervasive role in modern commerce and finance. A benchmark is a summary statistic over market variables, such as prices of specified assets at designated times. Benchmarks are employed by market participants for various purposes, including as reference measures for asset values (e.g., the S&P 500),

interest rates (LIBOR), and market volatility (VIX); to define derivative instruments; or as price terms in contracts (Gellasch and Nagy, 2019). Benchmarks in the form of reference measures can provide a concise reflection of market realities, thereby supporting decision making in the real economy. As such, accurate benchmark prices constitute a positive externality from functional financial markets (Bond et al., 2012). Their use in financial instruments and contracts also serve a valuable function in commerce and risk management.

Given their role in market decisions and contracts, some entities may have strong stakes in benchmark values, and hence incentives to try to influence or *manipulate* them. For instance, the *London Interbank Offered Rate* (LIBOR), an estimate of the rate at which banks can borrow from each other, supports more than $300 trillion worth of loans around the world (McBride, 2016). Several major banks have been implicated in schemes to manipulate LIBOR in the last decade, and criminal charges have been brought against over twenty individuals in the U.S. and U.K. since 2015 (McBride, 2016). February 2018 saw accusations of manipulation in the Chicago Board Options Exchange (CBOE) *Volatility Index* (VIX), a measure of U.S. stock market volatility based on the cost of buying certain options (Banerji, 2018). LIBOR has been particularly vulnerable to manipulation because it is calculated using self-reported data provided by parties with conflicts of interest regarding the benchmark's value (Duffie and Dworczak, 2018; Gellasch and Nagy, 2019). In the wake of the LIBOR scandal, regulators, academics, and market participants lobbied for a transaction-based replacement for LIBOR, such as the Secured Overnight Finance Rate (SOFR) or the U.S. Dollar Intercontinental Exchange (ICE) Bank Yield Index (Duffie and Dworczak, 2018; ICE Benchmark Administration Limited, 2019). Whereas it may be harder to manipulate transaction-based benchmarks, it is still possible, as in the alleged manipulation of the VIX in 2018 and the World Markets/Reuters Closing Spot Rates (WM/R FX rates) in 2014 (Boyle, 2014).

I introduce an agent-based model to shed light on how benchmark manipulation can operate, with the ultimate goal of supporting the design of manipulation-resistant benchmarks. My market model consists of a benchmark defined by the trade prices of a financial asset, which is traded on a continuous double auction (CDA). Trading agents only submit *limit-orders* to this CDA, which are orders to buy or sell at specified price or better. In this scenario it is possible for a market participant to trade on the CDA as well as be party to a contract whose value is tied to the benchmark. I refer to any profits made through a contract tied to the benchmark as *external holdings*.

I demonstrate how a party with external holdings can manipulate this benchmark through trading in the CDA. The first type of benchmark manipulator extends the behavior of the zero intelligence trader outlined in Section 3.2.1, adjusting its offers systematically in order to influence the benchmark in a certain direction. For example, if the manipulator wants to lower the benchmark, the manipulator can sell at a lower price to shift the benchmark down. If the manipulator submits orders priced lower than their valuation of the asset, then they may take a loss in their market profit. Though the manipulator may still earn a net profit if it successfully shifts the benchmark enough to impact its external holdings linked to the benchmark.

The second and third benchmark manipulators generates their trading strategies through deep reinforcement learning (DRL). I develop agents which try to learn a manipulative strategy with DRL by maximizing their rewards. The *reward* is the combined of profit from the market and external holdings given the state of the market and the action taken. The total profit of the manipulator is sum of its rewards over the trading period. I utilize DRL to maximize the manipulator's combined utility from the market and benchmark. I explore two DRL algorithms to train an agent with external benchmark holdings. The first is *deep Q-network* (DQN) (Mnih et al., 2015), an algorithm which observes a continuous environment and selects a discrete action. DQN uses a deep neural network to learn a value function over state-action pairs. The value function predicts the long-term payoff of the manipulator if it selects an action given the current state. The agent which uses DQN to learn trading strategies has the same action space as the ZI manipulator. The second algorithm is *deep deterministic policy gradient* (DDPG) (Lillicrap et al., 2016), which observes a continuous environment then selects an action from a continuous range. DDPG uses an actor-critic method where the critic learns a value function to inform the actor on its parameter selection. The actor then learns a distribution over the action space, which is a price range in this market environment.

An agent using DDPG to learn training strategies is the only manipulator that does not rely on ZI trading strategies. I design the reward function for both DRL agents to incorporate the agent's profits from the market and benchmark. This may cause the agent to submit misleading orders in an attempt to shift the benchmark to maximize its total profit.

Prior work has employed theoretical models and historical data to study benchmark manipulation in financial markets (Bariviera et al., 2016; Duffie and Dworczak, 2018; Duffie, 2018; Eisl et al., 2017; Rauch et al., 2013). Using a simulated market allows

us to incorporate complex details of *market microstructure*, representing the actual mechanics of trade, interactions among market participants, and the structure of the market. By combining the agent-based model with game-theoretic reasoning, I can also consider the response of strategic agents to the presence of a benchmark manipulator, and consider a wide range of market settings, benchmark designs, and trading strategy options.

I employ the standard market mechanism, discussed in Chapter III, organized around a limit order book for a single security. I assume a benchmark defined by transaction prices on this security. Trading agents may submit buy and sell orders, with orders executing with zero delay when matched, or resting in the order book pending execution against a subsequent order. The market is populated by a single manipulator, along with background agents who have private reasons to trade the security but no interests dependent on the benchmark. The manipulation activity potentially impacts the background agents through the market for this security, as well as (unmodeled) external parties who do not participate in the market but do have interests dependent on the benchmark.

I determine the impact of zero-intelligence benchmark manipulation by comparing strategic equilibria when the manipulator chooses to manipulate and to not manipulate. I also examine the effects of DRL manipulation by training the manipulator in a market with background traders and a market maker. I find in particular settings that manipulation is profitable overall to the manipulators. The manipulation activity itself is costly, in that the manipulator must take trading losses to move the benchmark. The background traders actually benefit from the manipulation, as their aggregate gains from trading increase. The external parties dependent on the opposite side of the benchmark are the real losers from the manipulation, with their losses captured in part by the manipulator and in part by the background agents whose trading is effectively subsidized.

This chapter is organized as follows. Following a discussion of related work in the next section, I describe the market environment in Section 5.3. Section 5.4 discusses the benchmark manipulator and the trading strategies it employs in this paper. In Section 5.5, I discuss the experimental design of my analysis. Section 5.6 presents the results with and without benchmark manipulation. I conclude this chapter in Section 5.7.

5.2 Related Work

The majority of prior work on benchmark manipulation is either theoretical or based on analysis of historical market data. Duffie and Dworczak (2018) introduce a theoretical model to analyze the robustness and bias of alternative benchmark constructions, and find that *volume-weighted average price* (VWAP) is optimal among linear benchmarks. Duffie (2018) also considers robustness to manipulation in design of an auction mechanism to convert LIBOR-based contracts to employ the replacement SOFR benchmark.

Bariviera et al. (2016) and Eisl et al. (2017) use historical data to find instances of manipulation of interest-rate benchmarks and provide suggestions for more robust benchmarks and regulation. Rauch et al. (2013) also use historical data to find instances of benchmark manipulation in LIBOR and investigate which banks were potentially involved in the 2011 scandal. Griffin and Shams (2018) examine spikes at time of settlement as evidence for possible manipulation of the VIX benchmark. Such findings have underscored concerns and contributed to policy discussions around reforms of financial benchmarks (Duffie and Stein, 2015; Gellasch and Nagy, 2019; IOSCO, 2013; Verstein, 2015).

There exists a significant amount of prior work that focuses on the goal of designing dynamic, successful trading strategies by utilizing various forms of reinforcement learning (RL). Previous studies address this in agent-based simulation and with historical data. Numerous studies develop trading agents within a simulation that learn profitable trading studies from a discrete observation and action space (Rummery and Niranjan, 1994; Schvartzman and Wellman, 2009; Sherstov and Stone, 2004; Wright and Wellman, 2018). Most recently, Amrouni et al. (2021) implements an agent that learns its trading strategy with DRL from a continuous observation space and discrete action space in ABIDES, a simulated market platform outlined in Chapter III. A similar patterns appears with exploring trading strategies learned through RL and historical data. A few prior studies use historical data to learn a trading strategy with a discrete observation and action space (Moody et al., 1998; Nan et al., 2020; Nevmyvaka et al., 2006). The majority of prior work on training trading strategies with historical data utilize DRL with a continuous observation space and discrete action space (Deng et al., 2017; Li et al., 2019; Théate and Ernst, 2020; Wu et al., 2020; Zhang et al., 2020). Lastly, the most recent work on trading strategy generation with DRL and historical data considers continuous observation and action spaces (Liu

et al., 2020; Ponomarev et al., 2019; Wu et al., 2020; Xiong et al., 2018; Yang et al., 2020). While these last studies use DRL algorithms that learn to select actions over a continuous action space, each of these studies reduce their trading strategy to a discrete action space. The prominent pattern throughout these prior studies is that as RL and DRL algorithms improve, they are applied to trading to find profitable strategies.

To my knowledge this is the first work to study financial benchmark manipulation in an agent-based model and analyze this type of manipulation's impact on market welfare. Analyzing benchmark manipulation in a simulated environment allows a study on market interactions to this type of manipulation. I explore benchmark manipulation through multiple trading strategies, including some generated through two DRL algorithms. Both utilize a continuous observation space, though one has a discrete action space while the other a continuous action space.

5.3 Market Environment

I study financial benchmark manipulation in an agent-based market model. This market consists of single asset traded on a CDA and a transaction-based benchmark calculated at the end of the trading period. I extend my market model by introducing a transaction-based benchmark. I also introduce two new trading strategies which aim to manipulate the market benchmark. This market is simulated in Market-Sim, a market simulation platform discussed in more detail in Chapter III.

5.3.1 Benchmark

After the termination of the market at time T, the benchmark β_T is calculated. There are numerous ways to define a benchmark, but in this study I use *volume-weighted average price* (VWAP). A benchmark calculated with VWAP sums the prices weighted by quantity of all transactions over the trading horizon. More formally, a VWAP benchmark:

$$\beta_T = \frac{\sum_{i=1}^{N} q_i p_i}{\sum_{i=1}^{N} q_i},$$

where q_i and p_i are the quantity and price of the i^{th} transaction. Given that agents submit only single-unit orders in this market, the VWAP benchmark becomes:

$$\beta_T = \frac{\sum_{i=1}^{N} p_i}{N}.$$

VWAP is representative of many benchmarks currently used or proposed for various purposes (Duffie and Dworczak, 2018).

5.3.2 Agents in the Market

My market is populated by numerous background agents and one benchmark manipulator. Section 3.2 describes the strategies background traders use in each market environment. The background traders use ZI trading strategies, and one environment contains a market maker. ZI agents and the market maker have no stake in the benchmark so it is not incorporated in their strategies.

5.4 Benchmark Manipulation Strategies

I model a benchmark manipulator that determines the value of the traded asset with an estimate of the common fundamental value and a private value. The manipulator uses the valuation model defined in Section 3.1. This agent is also party to a contract dependent on the benchmark, and receives a payoff from this contract directly proportional to the benchmark. I define the coefficient of agent's benchmark payoff as the *contract holdings*. Thus, the agent has incentive to influence the benchmark if it can raise its total profit between the market and benchmark.

I evaluate a benchmark manipulator who submits orders to a CDA like the background traders, and attempts to maximize its total profit between its market transactions and the benchmark. The total profit of a benchmark manipulator $B(t)$ at time t becomes:

$$B(t) = V(t) + \psi\beta_t, \tag{5.1}$$

where ψ is the agent's contract holdings and $V(t)$ is the valuation of the agent's market holdings, more formally defined in Equation 3.8. If $\psi < 0$, then the manipulator tries to lower the benchmark to maximize its total profit. Conversely, if $\psi > 0$, then the manipulator attempts to raise the benchmark. The manipulator must determine which market actions will maximize $B(T)$, its combined payoffs between the market and benchmark. If the manipulator wishes to lower the benchmark, then it can submit sell orders priced lower than its valuation of the asset. If these orders match with a buy order, then the resulting trade will be priced lower than if the manipulator submitted an order priced at or above its valuation of the asset. These lower priced transactions

will lower a VWAP benchmark. The reverse is true if the agent hopes to raise the benchmark; it can influence the price upward by submitting buy orders priced higher than its valuation. Thus, the manipulator must determine how to price its orders so it can influence the benchmark, but not lose too much from its actions in the market.

5.4.1 Manipulation with Zero-Intelligence

The first manipulation strategy I consider is *ZIM*, an adjusted version of a ZI strategy that attempts to shift the benchmark. A standard ZI agent submits orders priced at $p^{\mathbf{ZI}}(t) = v(t) \pm \zeta_t$, where ζ_t is the agent's desired profit. A ZIM agent offsets $p^{\mathbf{ZI}}(t)$ by $sign(\psi)\chi$, where $sign(\psi)$ is the sign of the trader's benchmark holdings and χ is the amount the manipulator decides to offset the price. More formally, a ZIM agent arriving at time t generates a limit price $p^{\mathbf{ZIM}}(t)$ by:

$$p^{\mathbf{ZIM}}(t) = p^{\mathbf{ZI}}(t) + sign(\psi)\chi. \tag{5.2}$$

This manipulator uses the strategic profit threshold parameter $\eta \in [0, 1]$ to consider the current visible quoted price. The application of η differs for a ZIM agent than that of a ZI agent. For a ZI agent, the desired profit is always such that $\zeta_t \geq 0$. However, this might not hold true for a ZIM agent if the benchmark offset parameter χ is large enough. If the ZIM agent's total desired profit $\zeta_t \pm sign(\psi)\chi < 0$, then the manipulator is willing to accept any portion of its desired profit, rather than just a fraction it like the ZI agent. If buying, the manipulator will price its order at the best available sell order ASK_t rather than $p^{\mathbf{ZIM}}(t)$ if:

$$\text{ASK}_t \leq v(t) + C_1, \tag{5.3}$$

where

$$C_1 = \max\Big\{\eta\Big(sign(\psi)\chi - \zeta_t\Big), \Big(sign(\psi)\chi - \zeta_t\Big)\Big\}.$$

If selling, it will price its order at the best available buy order BID_t rather than $p^{\mathbf{ZIM}}(t)$ if:

$$\text{BID}_t \geq v(t) + C_2, \tag{5.4}$$

where

$$C_2 = \min\Big\{\eta\Big(sign(\psi)\chi + \zeta_t\Big), \Big(sign(\psi)\chi + \zeta_t\Big)\Big\}.$$

5.4.2 Manipulation with Deep Reinforcement Learning

I also develop manipulative strategies using DRL. I deploy these manipulators in Market-Sim using the methods outlined in Section 3.3. The manipulator uses the DRL algorithms deep Q-network (DQN) and deep deterministic policy gradient (DDPG) to learn a trading strategies that maximizes its combined utility from the market and benchmark. I refer to the agents learning trading strategies through DQN and DDPG as the *DQN agent* and *DDPG agent*, respectively. DQN is a DRL algorithm which observes a continuous environment, then selects a discrete action (Mnih et al., 2015). DQN attempts to learn the optimal action given the current environment by updating a value function. DDPG is a DRL algorithm that uses a continuous observation and action spaces (Lillicrap et al., 2016). DDPG searches over a policy space in an attempt to find the optimal price given the current environment.

Deep Q-Network DQN is a model-free, off-policy value learning algorithm. It is *model-free* as it does not employ explicit estimates of state-action transition probabilities to update its policy. A *policy* defines the agent's behavior and is a mapping from states to actions. I deem the trading strategy of the agent to be the policy learned by the DRL algorithm. DQN is *off-policy* as it updates a target policy that is different from the policy used to generate experiences. It is imperative to use an off-policy algorithm in Market-Sim because I train the agent in between each market run, or the agent does not update its policy while the market is active. *Value learning* tries to learn a value function over state-action pairs.

DQN combines Q-learning and deep neural networks (DNNs) to learn Q-values in environments with rich sensory data. A *Q-value* is the estimated value of the total reward over the episode given the selected state-action pair, s and a respectively. The intermediate rewards are defined as ρ. Upon each market arrival, the agent records experience tuples $(s, a, s'\rho)$ to learn from and update its Q-values. DQN uses a DNN to learn a hierarchical abstract representation of a complex state space. This DNN estimates Q-values over a discrete action space. DQN updates the parameters θ of its network by following the stochastic gradient descent updating rule. The gradient descent rule of the network's parameters $\Delta\theta$ is defined as:

$$\Delta\theta = \alpha \left[(\rho + \gamma \max_{a'} Q_\theta(s', a')) - Q_\theta(s, a) \right] \nabla_\theta Q_\theta(s, a), \tag{5.5}$$

where $Q_\theta(s, a)$ is the estimated Q-value given the current network parameters and

state-action pair, α is the learning rate, and γ is the discount factor.

Deep Deterministic Policy Gradient DDPG is a model-free, off-policy actor-critic algorithm. An *actor-critic* algorithm combines policy learning and value learning. *Policy learning* tries to directly learn a policy function that maximizes the agent's reward. The actor represents a policy function and the critic represents a value function. The critic uses a Q-Network to inform the actor on its parameter selection. Then the actor uses the information from the critic to learn a distribution over the action space. The critic updates its Q-values through a DNN like DQN. The actor is updated given the learned parameters from the critic θ^Q, and by applying the chain rule to the expected return from the distribution J with respect to the parameters of the actor θ^μ. More formally, the actor's update rule is:

$$\begin{aligned} \Delta_{\theta^\mu} J &\approx \mathrm{E}_{s_t \sim \nu^\pi}\Big[\Delta_{\theta^\mu} Q(s, a \mid \theta^Q) \mid_{s=s_t, a=\mu(s_t|\theta^\mu)} \Big] \\ &= \mathrm{E}_{s_t \sim \nu^\pi}\Big[\Delta_a Q(s, a \mid \theta^Q) \mid_{s=s_t, a=\mu(s_t)} \Delta_{\theta^\mu} \mu(s \mid \theta^\mu) \mid_{s=s_t} \Big], \end{aligned} \tag{5.6}$$

where ν^π is the discounted state visitation distribution for a stochastic behavior policy π. A neural network is used to define the computation graph's architecture. This allows the algorithm to handle a large state space by learning a hierarchical abstract representation of the data. The actor learns a distribution over the action space, which is mapped to a continuous action space. Noise $\mathcal{N}$ is added to the actor's policy for exploration:

$$\mu'(s_t) = \mu(s_t \mid \theta_t^\mu) + \mathcal{N}. \tag{5.7}$$

State Space The benchmark manipulator's state space includes all the agent's private information. It also includes any published market and benchmark information. Appendix B provides a table of detailed descriptions of the features included in the agent's state space.

The agent considers multiple features in the state space that are private information. This includes its private valuation of the traded asset, and its current holdings of the asset and benchmark. I also include the order *side*, which defines whether the order is a buy or sell order. The agent chooses the side before each trade by flipping a coin to determine if it will buy or sell.

The agent's state space also includes publicly available information in the market,

such as the remaining number of time steps in the trading period and the number of time steps since the last trade. I also include features from the market's order book, such as the size, spread and currently listed order prices. The state must be a constant size, but the order book is dynamic throughout the trading period. I address this problem by specifying a limited fixed-size depth of book. I treat this fixed-depth as a hyperparameter. Then I pad or truncate the fixed-depth order book. The padded prices are an estimate of the final fundamental plus or minus three standard deviations of the observation noise. The bid padding price:

$$\hat{r}_t - 3\sigma_n.$$

The ask padding price:

$$\hat{r}_t + 3\sigma_n.$$

Padding is less necessary in the environments where a market marker is present. I also include the omega ratio, a metric that determines the favorability of submitting an order. Lastly, I include the number of transactions and their prices. The transaction price history must also have a set length, so I also pad or truncate this vector. I use an estimate of the final fundamental as the padded price.

The state space includes no private information from other agents. It also excludes which parties participated in each transaction. I omit this information because it is not accessible to any market participant in a trading game.

Action Space The benchmark manipulator's learned policy selects the price of the order to submit. Upon each arrival, the manipulator submits an order. It determines whether to buy or sell by flipping a coin, then it includes the side in the state space. My agent only submits single-unit orders. I do not give the manipulator the option to select no action and not submit an order. The agent can achieve the same goal though by submitting noncompetitive orders. A very low buy order and a very high priced sell order are noncompetitive orders.

When the benchmark manipulator learns a policy through DQN, its action space is to select from a variety of ZIM strategies. DQN maps to a discrete action space, so upon every arrival the agent's policy estimates which action will yield the highest long-term reward given the current state of the market.

When the benchmark manipulator uses a policy learned through DDPG to select an action, it directly selects a value $A \in [0, 1]$. My agent then maps this action to a

price at time t:

$$p_t^{\mathbf{DDPG}} = \hat{r}_t + \Big(C - sign(\psi)\chi\Big)A, \tag{5.8}$$

Where C is some constant treated as a hyperparameter during training, $sign(\psi)$ is the direction of the agent's contract holdings, and χ is an offset parameter. This mapping function is very similar to Equation 5.2, though rather than randomly selecting a desired profit from a uniform distribution, the agent learns the desired profit directly. After the agent calculates the price, it submits its order containing the price, side, and size.

Reward Function The benchmark manipulator designs its reward function to maximize its profits from both the market and benchmark. I define an agent's current profit from the market and benchmark $B(t)$ from Equation 5.1.

I define the agent's reward for the action taken at time t as the difference between the total profit at its next arrival at time t' and the total profit at time t:

$$\rho_t = B(t') - B(t).$$

I capture the total realized profit from the agent's action at time t by calculating the reward as the difference between the total profit at the next arrival and current arrival. In a market environment the reward from an action is not immediately known. The order placed at time t can match with another anytime between t and t'. The manipulator also cancels any active orders upon its next arrival. Thus, if the agent waits to calculate its reward until its next arrival, it guarantees to incorporate any reward from a transaction.

At the end of the market at time T, the summation of the rewards is equivalent to the manipulator's final payoff: market and benchmark:

$$B(t) = \sum_{t=\tau_1}^{\tau_n} \rho_t, \text{ s.t. } t \in \{\tau_1, \ldots, \tau_n\}.$$

Where τ_j corresponds to the time of the agent's jth arrival to the market.

5.5 Experiments

An agent using one of the benchmark manipulation strategies can likely shift the benchmark and increase its own total profit, but this may have negative ramifications on market welfare. I examine the impacts of these manipulative strategies in multiple market environments, employing a variety of strategies for the background agents and benchmark manipulator. I then find the combination of strategies that background traders play in equilibrium in the presence of manipulation. To determine the impact of benchmark manipulation, I find the payoffs of the manipulator, background agents, and aggregate market.

5.5.1 Market Environment Settings

I evaluate the benchmark manipulation strategies and their impact on market welfare in various market environments. Each environment has ten or fifteen background agents and one benchmark manipulator. I split the environments into sets **A** and **B**. Environment **A** is similar to that of Wang et al. (2018). The ZIM strategy was initially tested in the environments with Environment **A**. I use EGTA on Environment **A** to find the equilibrium strategies for each agent in the market. The environments within Environment **A** are too volatile to train an agent that learns its trading strategy through DRL, so this manipulative strategy is tested in Environment **B**. Environment **B** is the same as the market environment presented by Wright and Wellman (2018). I use the pure strategy equilibrium found by Wright and Wellman (2018) as an initial strategy profile for this environment then test pure strategy deviations, as outlined in Section 3.4.2.

In Environment **A**, the benchmark manipulator is assigned contract holdings $\psi \in \{-40, 40\}$. I denote settings where the manipulator possesses negative contract holdings and hopes to shift the benchmark down ($sign(\psi) = -1$) as ψ^-. Similarly, I label when the manipulator has positive contract holdings and hopes to shifts the benchmark up ($sign(\psi) = 1$) as ψ^+. In Environment **B**, the benchmark manipulator is assigned contract holdings $\psi = 40$.

Environment A The market is populated with ten background agents and one manipulator. The global fundamental time series of the market is produced by Equation 3.1, with fundamental mean $\bar{r} = 10^6$ and mean reversion $\kappa = 0.05$. The finite time horizon of the market is $T = 10,000$ time steps. All agents arrive to the market

Table 5.1: Environments within Environment **A**.

Environment	**LSHN**	**MSMN**	**HSLN**
σ_s^2	10^5	5×10^5	10^6
σ_n^2	10^9	10^6	10^3

Table 5.2: Strategies employed by the background traders (**ZI**) in Environment **A**.

Strategy	$\mathbf{ZI_1}$	$\mathbf{ZI_2}$	$\mathbf{ZI_3}$	$\mathbf{ZI_4}$	$\mathbf{ZI_5}$
$R_{\min}$	0	0	0	0	0
$R_{\max}$	1000	1000	1000	1000	1000
η	0.1	0.3	0.5	0.8	1.0

according to a Poisson distribution with rate $\lambda_a = 0.005$. The maximum number of units an agent can hold at any time is $q_{\max} = 10$. Lastly, the private value variance is $\sigma_{PV}^2 = 5 \times 10^7$.

There are three environments within the set Environment **A**. Theses environments differ in market shock σ_s^2 and observation noise σ_n^2. The market shock dictates fluctuations to the true value of the fundamental, so higher shock variance tends to produce higher price volatility. The observation noise controls the accuracy of agents' information on the fundamental. Thus, higher observation variance leads to less accurate fundamental information for agents. Table 5.1 presents the three environments where naming convention refers to low (L), medium (M), and high (H) market shock (S) and observation noise (N).

Each agent selects a strategy to play during a market run. The background agents select from the strategies presented in Table 5.2. The background agents choose between five ZI strategies. These five strategies have a fixed $R_{\min}$ and $R_{\max}$ values, and the background agents search over various η values. I included more ZI strategies in preliminary tests, but this is the set of strategies the background traders selected in equilibrium. The ZIM agent chooses from the strategies listed in Table 5.3. The strategy set of the ZIM agent explores varying benchmark impact parameter χ and the same $R_{\min}, R_{\max}$, and η values as the $\mathbf{ZI}_4$. The ZIM agent has the option to not manipulate the benchmark by choosing the $\mathbf{ZIM}_1$ strategy. I determine which χ value maximizes the ZIM agent's profit in each environment.

Table 5.3: Strategies employed by the ZI benchmark manipulator (**ZIM**) in Environment **A**.

Strategy	$\mathbf{ZIM_1}$	$\mathbf{ZIM_2}$	$\mathbf{ZIM_3}$	$\mathbf{ZIM_4}$	$\mathbf{ZIM_5}$
$R_{\min}$	0	0	0	0	0
$R_{\max}$	1000	1000	1000	1000	1000
η	0.8	0.8	0.8	0.8	0.8
χ	0	500	1000	2500	5000

Environment B The market environments within Environment **B** are populated with fifteen background agents, one market maker, and one manipulator. The changes in the parameter selection of the environment and the inclusion of a market maker reduce the overall noise between the environments in Environment **B** compared to those in Environment **A**. I find that reducing the overall noise in the market improves the ability of the DRL algorithms to learn successful trading strategies. The global fundamental time series of the market has a fundamental mean of $\bar{r} = 10^5$, mean reversion $\kappa = 0.01$, and a market shock variance $\sigma_s = 2 \times 10^4$. No noise is added to agents' observation of the fundamental, so $\sigma_n = 0$, which makes it unnecessary to consider transaction prices like a ZIT agent. Therefore, I only consider background traders that use ZI strategies. The finite time horizon of the market is $T = 2,000$ time steps. The background agents and manipulator arrive to the market according to a Poisson distribution with rate $\lambda_a = 0.012$. I consider instances of this market with and without a market maker. If the MM is present, the market maker also arrives to the market according to a Poisson distribution, but with rate of $\lambda_{mm} = 0.05$. The maximum number of units all agents can hold at any time is $q_{\max} = 10$. Lastly, the private value variance is $\sigma^2_{PV} = 2 \times 10^7$.

Table 5.2 specifies the strategies used by the background traders, which are the same as those employed by Wright and Wellman (2018). Table 5.3 lists the strategies of the ZIM benchmark manipulator. The ZIM agent chooses between the $\mathbf{ZI_7}$ and $\mathbf{ZI_8}$ strategies with varying benchmark impact χ. The ZIM agent has the option to not the manipulate the benchmark with the $\mathbf{ZIM_1}$ and $\mathbf{ZIM_5}$ strategies. I explore varying χ values to determine if it is profitable for the ZIM agent to manipulate the benchmark. More ZIM strategies were explored, but they were not selected by during single-player deviations.

I also examine environments where the manipulator learns trading strategies with

Table 5.4: Strategies employed by the background traders (**ZI**) in the low variance environment.

Strategy	$\mathbf{ZI_1}$	$\mathbf{ZI_2}$	$\mathbf{ZI_3}$	$\mathbf{ZI_4}$	$\mathbf{ZI_5}$	$\mathbf{ZI_6}$	$\mathbf{ZI_7}$	$\mathbf{ZI_8}$	$\mathbf{ZI_9}$	$\mathbf{ZI_{10}}$
$R_{\min}$	0	0	90	140	190	280	380	380	460	950
$R_{\max}$	450	600	110	210	210	380	420	420	540	1050
η	0.5	0.5	0.5	0.5	0.5	0.5	0.5	1.0	0.5	0.5

Table 5.5: Strategies employed by the ZI benchmark manipulator (**ZIM**) in the low variance environments.

Strategy	$\mathbf{ZIM_1}$	$\mathbf{ZIM_2}$	$\mathbf{ZIM_3}$	$\mathbf{ZIM_4}$	$\mathbf{ZIM_5}$	$\mathbf{ZIM_6}$	$\mathbf{ZIM_7}$	$\mathbf{ZIM_8}$
$R_{\min}$	380	380	380	380	380	380	380	380
$R_{\max}$	420	420	420	420	420	420	420	420
η	0.5	0.5	0.5	0.5	1.0	1.0	1.0	1.0
χ	0	250	500	750	0	250	500	750

DQN and DDPG. The market maker submits 100 buy orders and 100 sell orders at each market arrival. The spread the market maker uses is $S = 1024$ and each order is spaced by $\xi = 100$. The market maker is not a player in this market game because its strategic parameters are fixed.

Appendix C presents the hyperparameters selected for DQN and DDPG.

5.5.2 EGTA Process

Environment A I model my market as a role-symmetric game, which consists of players divided into roles that each have a designated strategy set. My market splits players into two roles: background traders and a single benchmark manipulator.

I use deviation-preserving reduction (DPR) to estimate a Nash equilibrium for each market game in Environment A. DPR is discussed in more detail in Section 3.4.1. My reduced game becomes three background traders and one benchmark manipulator. In this setting one background agent deviates to a new strategy while the other nine background agents are further reduced to three. I sample at least 50,000 simulation runs for a specified strategy profile of each game to reduce sampling error resulting from stochastic market features.

The equilibria of each environment are presented in Appendix D. My benchmark manipulator explores five strategies, but the manipulator chooses strategy $\mathbf{ZIM_4}$ in

an equilibrium of each market setting I explore in this work.

Environment B I model the market as a role-symmetric game and split this game into two roles: background traders and a single benchmark manipulator. The initial strategy explored is the pure strategy equilibrium identified by Wright and Wellman (2018) in their original study of this environment. I then replace one of the background traders in this game with a ZIM, DQN, or DDPG agent. To analyze the performance of the ZIM agent, I try each ZIM candidate from Table 5.5 against the background agents' initial strategy selection. I train the DQN and DDPG agents separately against the background agents' initial strategy selection. Whichever strategy the manipulator selects, I then test single-player deviations for the background agents from Table 5.4. The simplified EGTA method I use to study a game with a combination of players using DRL-derived and heuristic strategies is presented in Section 3.4.2. A detailed description of the strategy profiles chosen during this process are presented in Appendix D.

5.6 Results

I examine how benchmark manipulation affects market welfare. In Environment **A**, I use EGTA to equilibrate each market game, then I analyze the impact of benchmark manipulation in each equilibrated market. I Environment **B**, some benchmark manipulators learning trading strategies through DRL. I also use single-player deviations to find market games to examine the impact of benchmark manipulation in Environment **B**. Specifically, I calculate the market profit and total profit of the benchmark manipulator where total profit aggregates the profit from market trading (i.e., market profit) and profit from the benchmark holdings. I also find the profit of the background traders. The total profit and market profit are the same for the background traders because they are indifferent to the final benchmark calculation. Lastly, I study the aggregate market profit and aggregate total profit. The aggregate market profit is the summation of the background traders' profit and the benchmark manipulator's market profit. The aggregate total profit of the system, which I define as the sum of the background traders' aggregate profit and the benchmark manipulator's total profit.

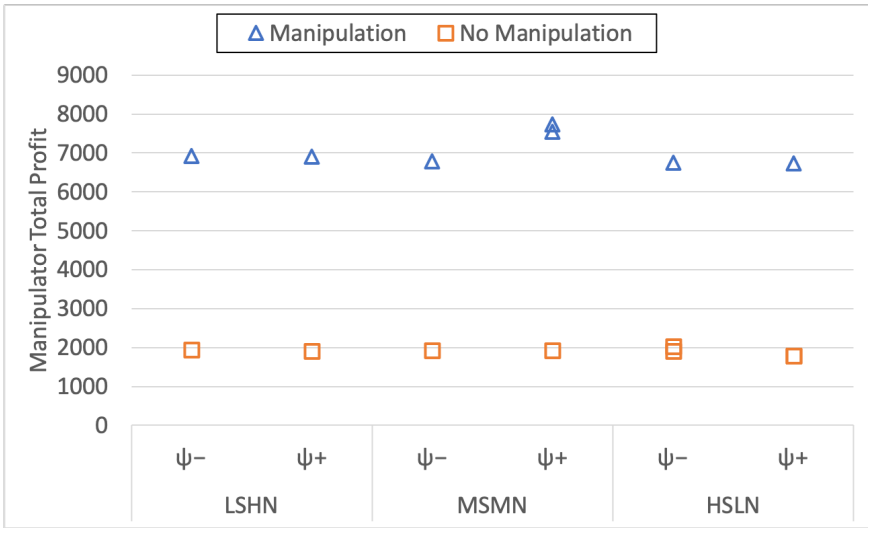
5.6.1 Environment A

In Environment **A**, I explore how a ZIM agent impacts the market welfare in an equilibrated market scenarios. I present the payoffs of the manipulator, background agents, and the aggregated market.

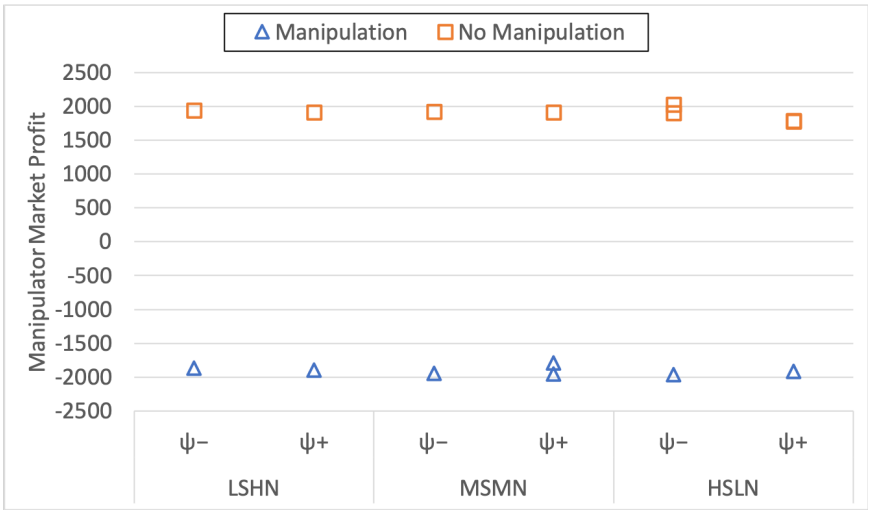
Figure 5.1 depicts the total profit and market profit of the benchmark manipulator, respectively. The total profit of the benchmark manipulator increases when it manipulates the benchmark. Of course, a profit-maximizing agent would not manipulate the benchmark if it did not increase its total payoff. The benchmark manipulator's market profit actually decreases when it manipulates the benchmark. This happens because in order to manipulate the benchmark, this agent must trade at prices it does not believe reflect its valuation. However, it is worthwhile to the agent to endure the decrease in market profit because its profits from the change in benchmark more than cover the loss.

Figure 5.2 shows the profit of the background agents with and without benchmark manipulation. The background agents benefit from benchmark manipulation. The manipulator takes a loss in the market to gain from an external contract dependent on the benchmark. If a background trader is on the other side of a transaction with the manipulator, then transaction is likely favorable for the background trader because the manipulator's market loss is the background agent's gain. Therefore, background traders' profit increases from matching with the manipulator's orders that are priced worse than the manipulator's valuation.

Figure 5.3 shows the aggregate total profit and aggregate market profit, respectively. The aggregated total profit, which I find by summing the total profit of the benchmark manipulator and background traders, increases with benchmark manipulation. The aggregated market profit I find by summing the market profit of the benchmark manipulator and background traders, and this value decreases with market manipulation. The allocation in the manipulated market is less efficient than in the market without manipulation, which is why the aggregated market profit decreases. Also, in equilibrium, this manipulation impacts the benchmark enough that the manipulator's gain from the benchmark exceeds its losses from trading in the market. The background traders gain at most the manipulator's loss from the market, but the manipulator's resulting gain from the external contract exceeds that of the background traders. Therefore, the counterparty to the manipulator in the benchmark contract loses precisely what the benchmark manipulator gains from the benchmark contract.



(a) Total (market plus benchmark-dependent contract) profit of the benchmark manipulator with and without manipulation.



(b) Market profit of the benchmark manipulator with and without manipulation.

Figure 5.1: Profit of the ZIM agent in Environment **A**. For both figures, the x-axis represents different market environments with varying fundamental shock, observation variance, and benchmark impact.

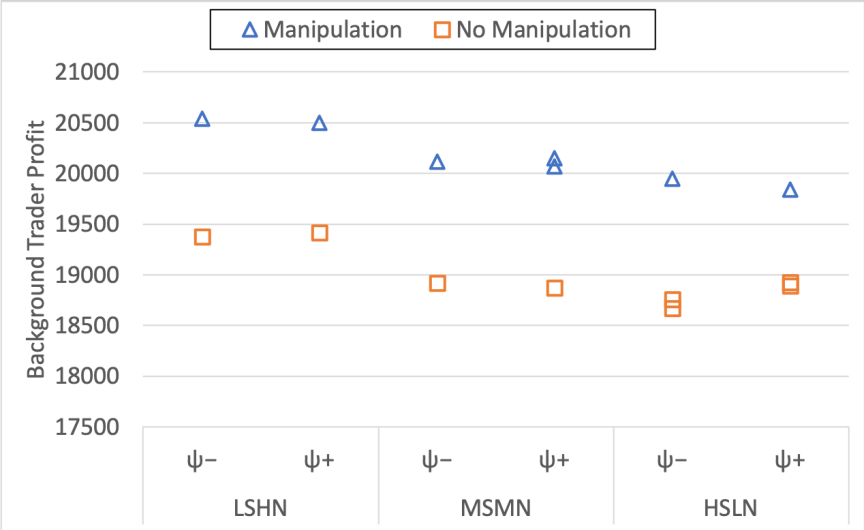
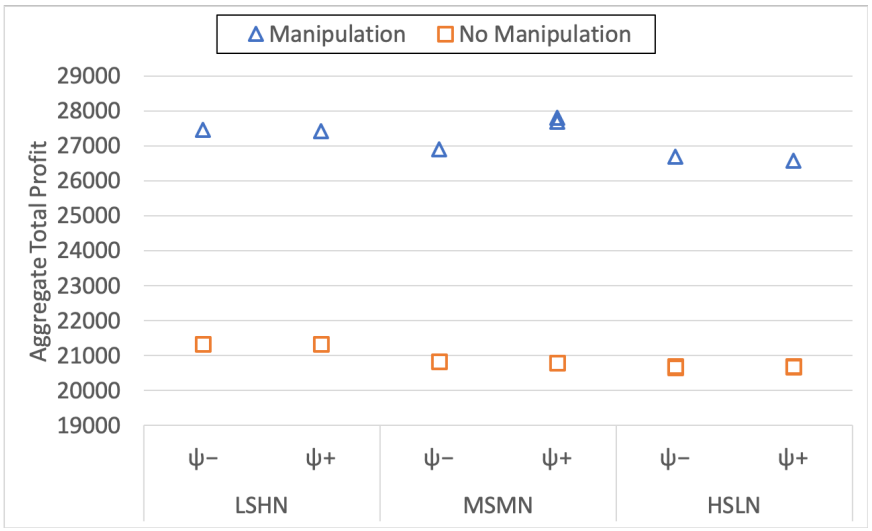


Figure 5.2: The aggregate total profit of the ten background agents with and without ZIM benchmark manipulation in Environment **A**. The total and market profit is the same for background agents. The x-axis represents different market environments with varying fundamental shock, observation variance, and benchmark impact.

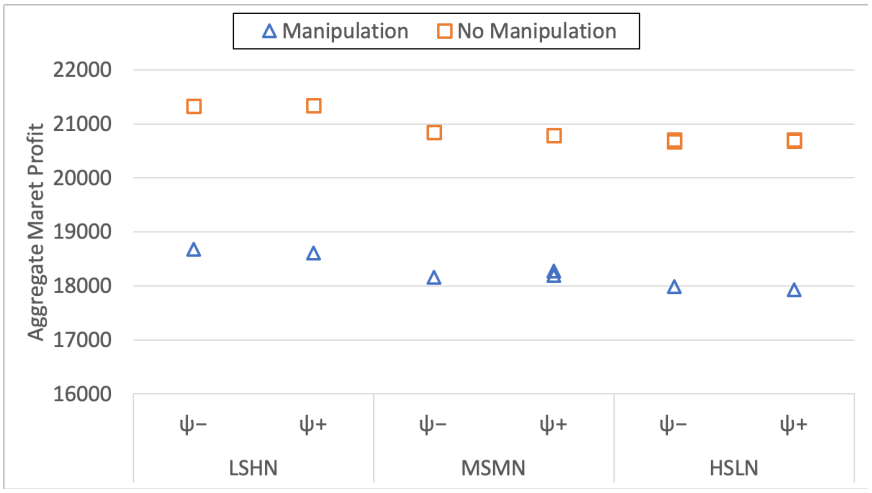
The analysis of this market outcome in isolation does not take into account the broader effects of benchmark manipulation. Manipulation reduces the usefulness of tethering the value of contracts to benchmarks. This manipulation may also lead to mispricing assets in the real and financial economy. For example, a mispriced LIBOR rate could result in banks lending to customers at rates that do not appropriately reflect their risk.

5.6.2 Environment B

I analyze the success of ZI, ZIM, DQN, and DDPG agents in Environment **B**. Each manipulator has contract holdings $\psi = 40$. Environment **B1** is the market environment where the background agents are equilibrated for no manipulation in a pure strategy equilibrium found by Wright and Wellman (2018). Environment **B2** refers to the market environment where the background agents are calibrated to the ZIM manipulator using the single-player deviation method. Environment **B3** denotes the market environment where the background agents are calibrated to the DQN manipulator using the single-player deviation method. When the manipulator is labeled "ZI," this signifies the case when it does not manipulate. I examine the training curve of the DQN agent and the valuation curve of the DDPG agent. I study



(a) Aggregate total profit with and without ZIM manipulation.



(b) Aggregate market profit of all agents with and without ZIM manipulation.

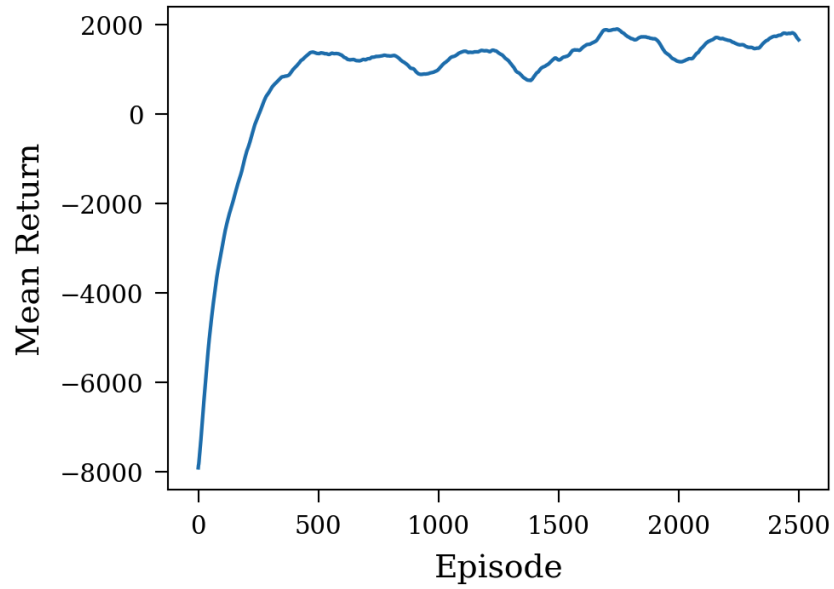
Figure 5.3: Total and market profit in Environment **A** with and without ZIM benchmark manipulation. In both figures, the x-axis represents different market environments with varying fundamental shock, observation variance, and benchmark impact.

the welfare impacts of the three manipulators by examining agent and aggregate market payoffs. If a MM is present, then its profit is also included in the aggregate total and market profits.

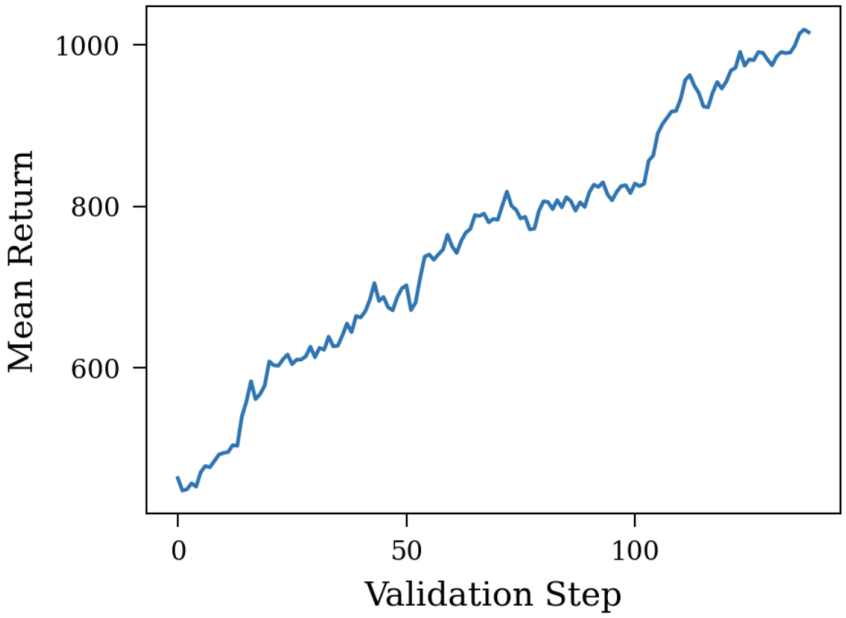
Figure 5.4 depicts the training and validation curves of the performance of the DQN and DDPG agents, respectively. The performance measure is an exponentially smoothed running average of the mean payoff over the last 100 episodes. Between training steps, experiences are collected to train with by running the market simulator using the current policy with exploration noise. The payoffs used to generate the DQN training curve are the manipulator's total payoff from these market runs. The DQN agent's mean return gradually increases over the trading period. DDPG adds a significant amount of noise to each action when training, so the training curve is less informative for this algorithm. Instead, I graph the validation curve, where periodically the DDPG agent tests its learned policy without adding noise. The validation curve gradually increases as the DDPG agent learns a policy. It's also possible that the DDPG agent needs to be trained longer and tested more frequently to reach a stable policy.

Fig. 5.5 depicts the total profit and market profit of the benchmark manipulator, respectively. In most cases, the total profit of the benchmark manipulator increases when it manipulates the benchmark. When a MM is present, the ZIM agent and DQN agent in **B2** and **B3** increased their average total profit from the non-manipulative case, but not by a significant amount. These manipulators' market profit decreases from the non-manipulative case. It is worthwhile for the successful manipulator to endure the decrease in market profit because its profits from the change in benchmark more than cover the loss. Also when a MM is present, the DQN agent in Environment **B1** and DDPG agent successfully increases its total profit and is the only strategy to do so by a significant amount compared to the non-manipulative agent. When a MM is not present all of the manipulators significantly increase their total profit and decrease their market profit. It is easier for the manipulator to increase its total profit when there is no MM because it does not need to trade through the MM's many orders in the book to change the price.

Figure 5.6 shows the profit of the background agents in Environment **B**. The background agents benefit from benchmark manipulation as the average payoff of the background agents increases for each manipulative agent compared to the non-manipulative agent. The background agents are significantly better off when there is no MM; this is likely because the manipulator trades with the background agents

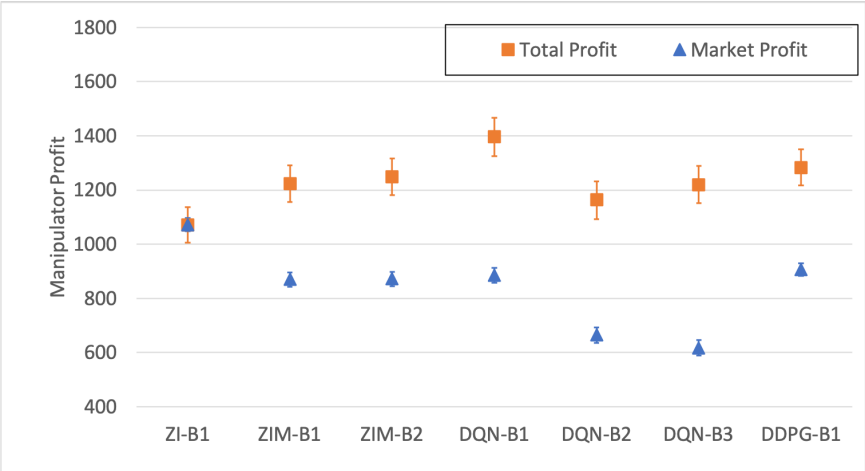


(a) Training curve of a DQN agent in environment **B1**.

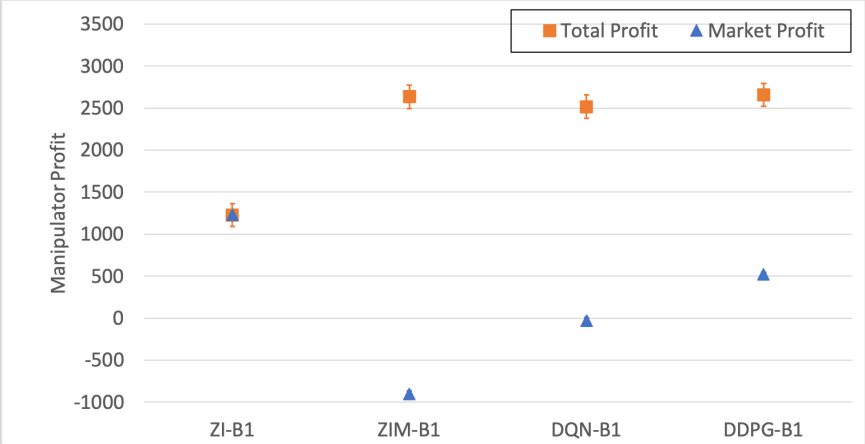


(b) Validation curve of a DDPG agent in environment **B1**.

Figure 5.4: The mean payoff of DQN and DDPG agents in training. This shows an exponentially smoothed running average payoff of the DQN and DDPG agents at each training step. The running average is over the previous 100 training steps.

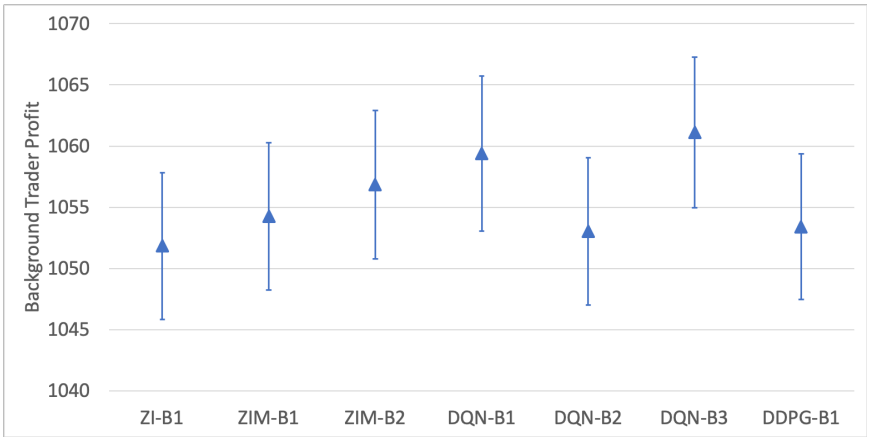


(a) MM present.

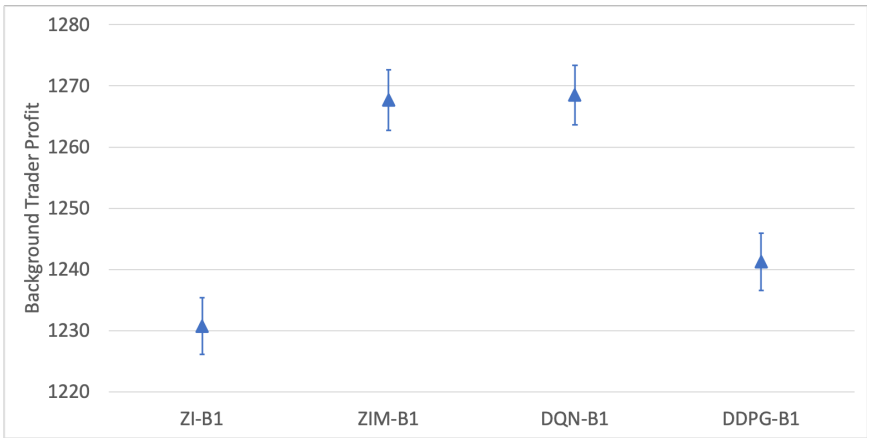


(b) No MM present.

Figure 5.5: Profit of the manipulator in Environment **B**. In both figures, the x-axis represents which strategy the manipulator uses and in which environment. Each point shows the average payoff of the manipulator with standard error bars.

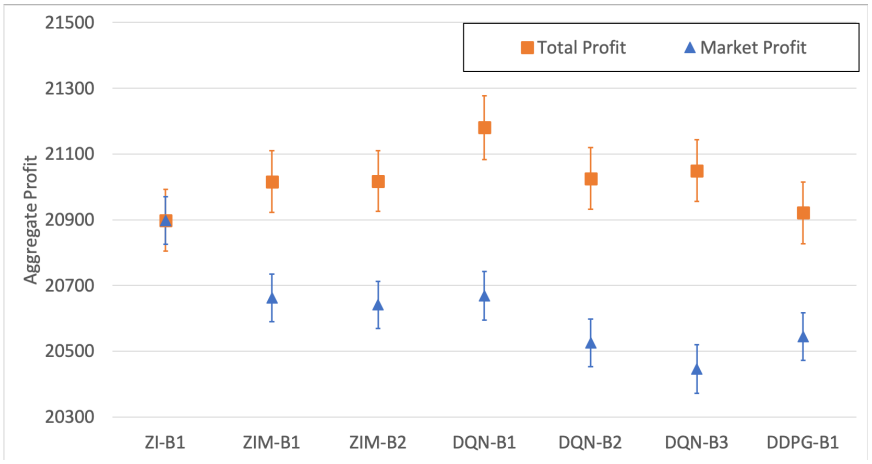


(a) MM present.

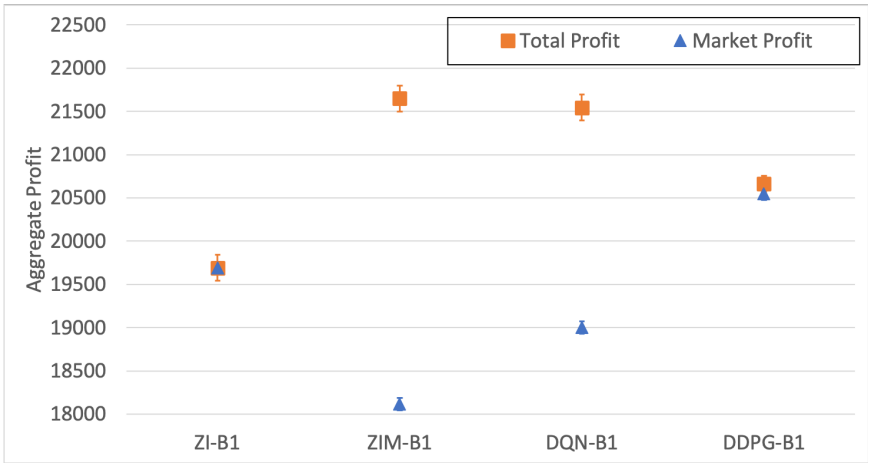


(b) No MM present.

Figure 5.6: The aggregate total profit of the fifteen background agents. The total and market profit is the same for background agents. In both figures, the x-axis represents which strategy the manipulator uses and in which environment. Each point shows the average aggregate background trader payoff with standard error bars.

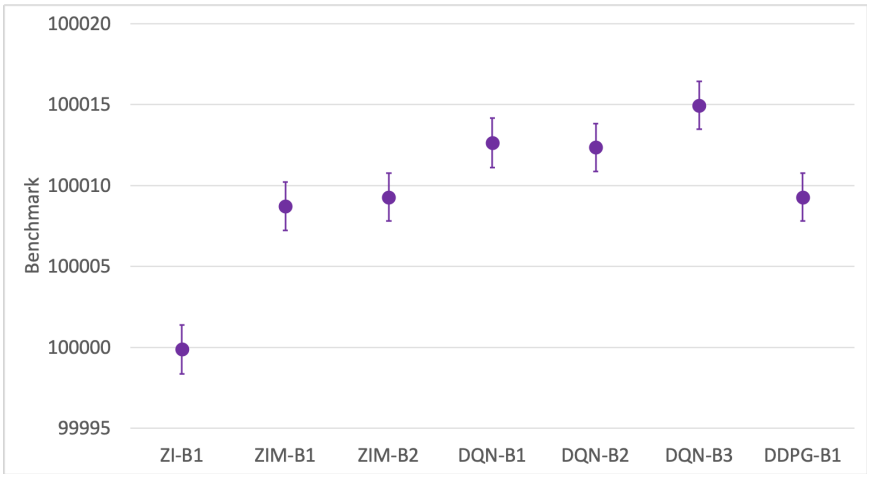


(a) MM present.

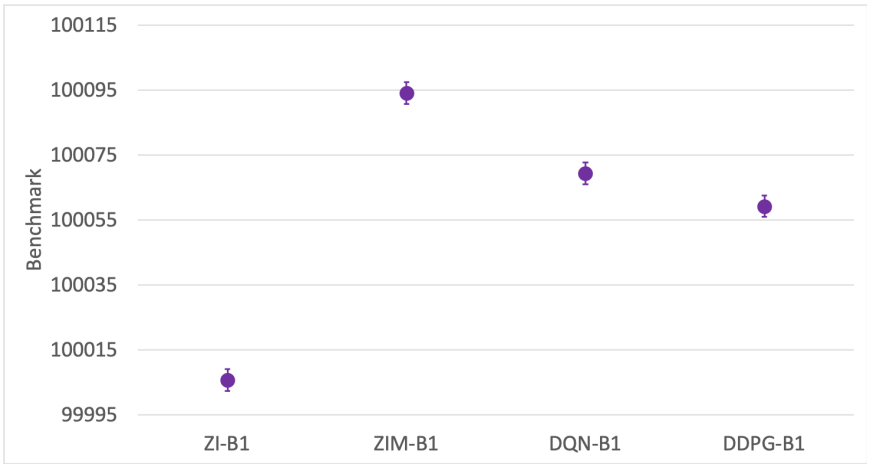


(b) No MM present.

Figure 5.7: Total and market profit in Environment **B**. In both figures, the x-axis represents which strategy the manipulator uses and in which environment. Each point shows the average aggregate payoff with standard error bars.



(a) MM present.



(b) No MM present.

Figure 5.8: The VWAP benchmark in Environment **B**. In both figures, the x-axis represents which strategy the manipulator uses and in which environment. Each point shows the average benchmark with standard error bars.

more in that case. The benchmark manipulator submits orders priced worse than their valuation of the asset to influence the benchmark. The background traders benefit from the manipulator's actions by matching with poorly priced orders. These poorly priced orders may increase the total number of transactions in the market because some of them will be priced so that they will match with another resting order. Background agents can also submit orders with higher demanded surplus that will be matched when the manipulators try to shift the benchmark, which will also increase their profit. Overall, the background agents benefit from benchmark manipulation.

Figure 5.7 shows the aggregate total profit and aggregate market profit, respectively. I find the aggregated total profit by summing the total profit of the benchmark manipulator and background traders. The aggregate total profit increases with benchmark manipulation. The aggregated market profit I find by summing the market profit of the benchmark manipulator and background traders. The aggregated total and market profits include that of the MM when one is present. The aggregate market payoff decreases with market manipulation. The market becomes less efficient when the manipulator is more successful. The least efficient market though is when the DQN agent does not have a successful trading strategy in Environment **B2**, in this case the manipulator loses and also significantly reduces the efficiency of the market. Benchmark manipulation impacts the benchmark enough that the manipulator's gain from the benchmark exceeds its losses from trading in the market. The background traders gain at most the manipulator's loss from the market, but the manipulator's resulting gain from the external contract exceeds that of the background traders. Therefore, the counterparty to the manipulator in the benchmark contract loses precisely what the benchmark manipulator gains from the benchmark contract.

Figure 5.8 depicts the VWAP benchmark in each market environment. The benchmark increases by a significant amount when there is manipulation compared to when there is no manipulation. The benchmark increases more when there is no MM present, though the manipulator is still able to successfully influence the benchmark in both cases. Therefore, the manipulator is able to successfully shift the benchmark in the direction of its contract holdings.

5.7 Conclusion

I analyze agents in a simulated market with a single traded asset to determine the impact of financial benchmark manipulation on market welfare. The financial

benchmark is calculated at the conclusion of a market by taking the volume weighted average price of all transactions executed during the market period. I design and implement three types of benchmark manipulators who wishes to shift the benchmark calculation to increase its profit from the benchmark, and multiple background traders who are indifferent to the benchmark. With or without MM, the manipulative activity increases profits of background traders, who thus have no incentives to help mitigate this type of manipulation. The manipulators that use DRL to generate trading strategies are more successful than the manipulator that uses a heuristic ZI-inspired strategy. I find that the profit of both the benchmark manipulator and background traders increases. Thus, background traders have no incentives to help mitigate this type of manipulation. Though the aggregate profit of the market participants increases, the aggregate total profit generated from the market decreases when the benchmark is manipulated, so the general welfare of the market decreases. Given that the profit of all market participants increases, third parties invested in the opposite direction of the manipulator bear a large portion of the manipulation costs.

The DQN and DDPG agents do not necessarily have an explicit intent to manipulate. Though they do learn manipulative trading strategies because of my algorithmic design choices. The DQN and DDPG agents attempt to maximize their profits, and the available action selections and incorporating benchmark profits in the reward function lead them to manipulative strategies. Current manipulation law in the US stock market requires an explicit intent to manipulate. Given the growing popularity and advances in DRL, it may be worth revisiting manipulation law to incorporate design choices for trading strategies learned through machine learning. Where a design choice which leads a manipulative strategy is considered manipulation.

A limitation of this study is that I consider only one simple benchmark calculation. Many financial benchmarks used to estimate asset values in real financial markets are derived from far more complex calculations. It is possible that these benchmarks are more (or less) difficult to manipulate than VWAP, so it would be worthwhile to explore the possibilities of manipulating benchmarks similar to them.

Another limitation of this study is that I only explore single player deviations in the low variance environment, rather than using EGTA to find an equilibrium. Ideally, I would equilibrate the background agents while a DQN or DDPG agent consistently uses a policy to select actions. Next, I would train the DQN or DDPG agent while the background agents play the strategies from in equilibrium. I would alternate those two steps until the background agents no longer chose a different strategy.

CHAPTER VI

Close Price Manipulation

6.1 Introduction

At the end of a trading day in the US stock market, each traded symbol finishes the day at a *close price*. The close price is frequently referred to as "the close." Contracts, derivatives, and reference prices for benchmarks commonly use the close as an evaluation metric (Alexander and Cumming, 2020). The close is also used to determine the net asset value of funds and company or fund performance evaluations (Comerton-Fordea and Putninš, 2011). Close price manipulation is difficult to study, but some believe it is common practice to manipulate the close, though rarely detected and prosecuted (Comerton-Fordea and Putninš, 2011).

It is challenging to build an accurate and comprehensive simulated market model to examine close price manipulation. Historical data from the US stock market provides a significant amount of useful information for analyzing this problem. Using historical data alone to determine the impact of market activity that did not actually happen provides several problems. Once a new trader enters the market and interacts with other traders, those other traders would change their behavior to reflect the new environment, so future trades and orders after the benchmark manipulator interferes become irrelevant and no longer viable.

I examine the impact of close price manipulation in a simulated market environment. This market model consists of a single asset that trades on two markets: a continuous double auction (CDA) and a close auction. A variety of agents trade on the CDA, including an agent that learns a trading strategy with the deep reinforcement learning (DRL) algorithm deep Q-network (DQN). The DQN agent is party to a contract dependent on the close price, so this agent tries to learn a trading strategy to maximize

its profits from the CDA and close price. I use a generative adversarial network (GAN) specifically designed to generate financial market orders (Li et al., 2020). I train this GAN on historical close auction data. This approach attempts to combine the benefits of data analysis and simulated market models.

In this initial study of incorporating historical data into simulation, I find that order streams generated by a GAN are not representative of real close auction orders. This likely happens because of the sensitivity of GANs and the small size of training dataset. GANs are sensitive to outliers and high variation in datasets (Quach, 2020), but the training set was difficult to process in a way that is compatible with a GAN without reducing the dataset size to a point of risking overfitting. A manipulator that learns trading strategies through DQN is not able to manipulate the close price in this market model. This likely occurs because the order stream generated by the GAN is not responsive to the manipulator's activity, and the manipulator would have to submit orders whose loss in the market would not outweigh the gain from shifting the benchmark.

This chapter is organized as follows. In Section 6.2, I examine prior work on close price manipulation. I discuss the US stock market data I use in Section 6.3. I discuss the simulated market model in Section 6.4. In Section 6.5, I present the learning agents that generate the synthetic order stream and attempt to manipulate the close. In Section 6.6, I provide the experimental setup. I present results in Section 6.7. Lastly, I conclude in Section 6.8.

6.2 Related Work

I adapt Stock-GAN from Li et al. (2020) to generate a stream of synthetic close auction orders trained from real data. Stock-GAN was initially used to generate orders in a CDA, so I make some adaptions for the close auction. In a study by Wang and Wellman (2020), a manipulator attempts to generate orders that are not recognizable as manipulative, and a discriminator attempts to distinguish between normal and manipulative trading activity. I use Stock-GAN in a different manner where I study how another agent attempts to manipulate the order stream generated by Stock-GAN.

The DRL manipulator is an adaptation of the manipulator which learns trading strategies through DQN in Chapter V. The close price can be thought of as a financial benchmark, so I treat it as such where I adapt the state space and reward function to close price manipulation.

Numerous prior studies used historical data and theoretical models to study close price manipulation. Comerton-Fordea and Putninš (2011) use historical data to analyze the impacts of close price manipulation in the US and Canadian stock markets. They found that when market participants manipulate the close, spreads widen and day-end returns increase. Kucukkocaoglu (2008) did a similar study on the Turkish stock market and found similar results. Hsieh (2015) analyzed data from the Taiwanese stock market and found that disclosing information five minutes before the close reduced manipulation. Frei and Mitra (2021) used an agent-based model to find the optimal close benchmark robust to manipulation, and determine that VWAP is this benchmark. Saakvitne (2016) developed a theoretical model to examine close price manipulation in foreign exchange markets, and concluded that this form of manipulation is natural and should not be considered manipulation.

6.3 Close Auction Order Data

In this chapter, I utilize real financial data to generate synthetic close auction orders with a Stock-GAN. The close auction for a stock is held at the exchange where the stock is listed. The majority of stocks are listed at the New York Stock Exchange (NYSE) or NASDAQ. NYSE still allows for price adjustments by humans, where NASDAQ does not. Therefore, NASDAQ is closer to being fully automated, so I aim to generate an order stream like that of NASDAQ. Given this nuance, I focus on NASDAQ's close auction to generate a simulated order stream for an agent-based model.

I use NASDAQ close auction orders, NASDAQ imbalance messages, and Securities Information Processor (SIP) quotes between 01 August 2018 and 26 October 2018. I also only consider NASDAQ listed stocks in the S&P 500. The close auction orders I consider are NASDAQ *limit-on-the-close* (LOC) orders, which are limit orders submitted to a close auction. I analyze orders from ten minutes prior to the close until the close, or between 15:50 and 16:00. My implementation of Stock-GAN only attempts to mimic the side and price of these LOC orders, so it does not attempt to find patterns in time and quantity. The *side* refers to whether an order is a buy or sell order. The orders are sequenced by arrival time to NASDAQ, and no LOC order can be modified or canceled after 15:50.

For each order, I record the most recent imbalance message[1]. During the examined

[1]https://www.nasdaqtrader.com/content/productsservices/Trading/ClosingCrossfaq.pdf.

time period, NASDAQ begins publishing imbalance messages for each symbol with a NASDAQ close auction at 15:50. The imbalance messages are published every five seconds until the market closes at 16:00. An imbalance message contains information relevant to the close auction, including price and quantity statistics of the close auction. Traders cannot view any order book information in the close auction like in the CDA, so imbalance messages provide pertinent information for traders who participate in the close auction. In my training dataset, I consider the imbalance message statistics: current reference price, near indicative clearing price, far indicative clearing price, and the imbalance side. The *current reference price* refers to the price within the best available bid and offer that the most shares would match at in the combined close auction and CDA. The *near indicative clearing price* is the price that will maximize the number of matched shares for the combined close auction and CDA. The near indicative clearing price is the best estimate for the final close price. The *far indicative clearing price* is the price that maximize the number of matched shares in only the close auction. Lastly, the *imbalance side* references the direction of the imbalance, or the side with more unmatched shares if the close auction was to close at the near indicative price.

I also record the most recent SIP quote on NASDAQ for each order. The training dataset I analyze includes the price of the best available bid and offer (BBO) on NASDAQ. This can differ from the current reference price in the most recent imbalance message if another order arrives at the continuous market and changes the BBO between the five second interval of imbalance message publication.

6.4 A Simulated Market with a Close Auction

I use the financial market simulator presented in Chapter III, Market-Sim. I extend Market-Sim by adding a close auction, which runs in parallel with a CDA. A single asset trades on the close auction and CDA. I model the close auction as a call market in the same manner as Wah and Wellman (2013). A *frequent call market* is a market where orders are matched at discrete, periodic intervals. At the clearing interval in a frequent call market, the market determines a single clearing price. The *clearing price* is a price that all buy orders above the price and all sell orders below the price are matched to transact at the determined price. The close auction is a call market where orders are not matched until the end of the trading period T. Even though the market is not cleared until the end of the trading period, agents can submit orders to

the close auction at any time during the trading period.

The close price is determined by finding the price at which the most shares in both the close auction and CDA will be matched. At the end of the trading day, any orders initially submitted to the CDA that can trade at the close price are matched in the close auction. So any active CDA orders are considered in determining the close price, by moving those active orders from the CDA to the close auction at time $T - 1$. Unlike the CDA, agents cannot modify or cancel orders in the close auction.

The close price is determined at the end of the trading period. The close price is set to the midpoint of the range of prices that would match supply and demand. Trades only execute if the supply and demand intersect, which occurs when the price of the best available bid is greater or equal to the best available ask. This is different than the NASDAQ close auction, which selects the price that maximizes the shares paired. I alter the close price calculation in an attempt to capture the missing volume in my simulated market. A NASDAQ close auction typically includes significantly more volume and order types than the market environment I model. Choosing the midpoint may capture some of the more competitive close orders which would have narrowed the *spread*, or the distance between best available bid and ask.

The close auction is more opaque than the CDA in that agents do not have access to any order book information. Instead, agents can view imbalance messages published at discrete, periodic intervals. I model four components of a NASDAQ imbalance message: near indicative price, far indicative price, reference price, and imbalance side. These metrics are similar to those discussed in Section 6.3 but are tailored to match the close price calculation in this market model. The near indicative price is what the close price would be considering all active orders in the close auction and CDA. The far indicative price is the what the close price would be only considering the active close auction orders. The reference price is the price between the CDA's best available bid and offer which the combined close auction and CDA would close. If it is not possible to calculate the near indicative price, far indicative price, or reference price because of a lack of order volume, then an estimate of the fundamental value is used in the place of any value which could not be calculated. The imbalance side is the side on which the most orders would be unmatched if the market closed at the near indicative price. If there is no imbalance, then the imbalance side is set to zero.

6.5 Learning Agents

In this chapter I deploy the Stock-GAN to learn a generative modal the produces a synthetic close auction order stream and a manipulator that learns trading strategies through DQN. The Stock-GAN trains from real NASDAQ close auction orders to generate a more realistic close auction order stream in the simulated market. The second is an agent that learns its trading strategies though DQN. The DQN agent submits orders to the CDA and attempts to manipulate the close price.

6.5.1 Close Auction Order Stream

The Stock-GAN consists of a generator and discriminator that compete against each other. The Stock-GAN considers a vector of features x_i, which is composed of the order price, order side, the CDA's best available bid, the CDA's best available ask, reference price, near indicative price, far indicative price, and imbalance side. Each feature vector is assumed to be independent and identically distributed. The generator is a model that aims to capture a realistic distribution $\mathcal{P}_r(x_i \mid x_{i-1}, \ldots, x_{i-k})$ of the close auction dataset. I do not allow any overlapping data sequences, i.e., if an order is included in the order history of one feature vector, it cannot be included in another. The *order history* is the k feature vectors that preceded the current feature vector. From the real dataset, the learned distribution of the generator model is $\mathcal{P}_g(x_i \mid x_{i-1}, \ldots, x_{i-k})$. The discriminator then calculates the Wasserstein distance between the two distributions. The overall goal is to minimize the distance between the two distributions.

I aim to approximate a realistic order stream in a simulated environment, rather than an agent which uses a strategic trading strategy to maximize its profit. The generator to select a price and side to submit as a close auction order. Synthetic close auction orders are submitted to the simulated market according to a Poisson distribution. The orders should be influenced by the imbalance messages and CDA best available bid and ask, so the DQN agent can potentially manipulate the synthetic close auction order stream, and therefore manipulate the close price.

6.5.2 Close Price Manipulation with Deep Reinforcement Learning

This agent extends the DQN manipulator in Chapter V. I extend this agent's state space by including features of imbalance messages. This extends the state space by

four features: reference price, near indicative price, far indicative price, and imbalance side. In this work, the manipulator is party to a contract dependent on the close price. Therefore, the manipulator's valuation of the asset is the same as the valuation defined in Equation 5.1. The benchmark β_t is set to the value of the close price. In the middle of the trading period, I use the near indicative price as an estimate for the close price.

6.6 Experiments

I use a market environment within Environment **B** presented in Section 5.5.1. The CDA in this environment consists of fifteen background traders using ZI strategies, one fundamental market maker, and one manipulator that uses a ZIM strategy or uses DQN to learn trading strategies. The manipulator has contract holdings of $\psi = 40$, so it attempts to increase the close price. The background agents play the strategies in the pure strategy equilibrium identified by Wright and Wellman (2018) in their original study of this environment. In this equilibrium the background traders play strategy $\mathbf{ZI}_6$ from Table 5.4. The ZIM agent uses the strategy $\mathbf{ZIM}_4$ from Table 5.5. Appendix E outlines the hyperparameters used while training DQN. I extend this environment by introducing a close auction and a generative model that produces a synthetic order stream of close auction orders. These synthetic orders are submitted the market according to a Poisson distribution with $\lambda_c = 0.024$. Like all other orders submitted to the market, synthetic close orders are single unit orders. The side and price are determined by the Stock-GAN's learned generator.

6.6.1 Stock-GAN Hyperparameters

I split the dataset of NASDAQ close auction orders into training and test sets. The full dataset consists of 124,674 orders. Each order consists of the price and side and is paired with the current CDA best bid and offer, reference price, near indicative price, far indicative price, and imbalance side. Each order is also conditioned on the previous 15 orders. An order can only appear once in the dataset, whether it's the primary order or part of an order's history. 80% of the dataset is reserved for training and the rest is included in the test set. I train the Stock-GAN for 50 epochs on a batch size of 768. An epoch is one iteration over all batches. At the beginning of epoch, the batches are resampled. A detailed summary of the generator and discriminator model for Stock-GAN is included in the Appendix F.

Table 6.1: MSE between the normalized true close auction orders in the test set and the normalized orders generated by Stock-GAN.

Order Component	Mean Squared Error
Price	0.022
Side	0.837

6.7 Results

This work studies whether a ZIM manipulator or DQN manipulator can shift the close price in a simulated market. The manipulator submits orders to the CDA, and the close auction order stream is synthetically generated by a generative model learned by a Stock-GAN trained on real market data. I analyze the quality of the close auction orders generated by the generative model. I also find the payoffs of the DQN manipulator and background agents to determine if the DQN agent is able to increase its payoff and manipulate the close.

6.7.1 Synthetic Close Auction Order Stream

The Stock-GAN is trained on a dataset of real close auction orders. A test dataset is reserved to test the quality of the orders the Stock-GAN generates compared to real close auction data. Table 6.1 presents the mean squared error (MSE) between a normalized real close auction order and a normalized synthetic order with the same order history. I find the MSE for the price and side, as other components in an order, such as size and time, are not predicted by the generative model in the simulated market. Given that the MSE error is found between the normalized orders, these values would have to be scaled back to the true prices. The side can only take on the values -1 and 1, but the price may be between \$0.01 and \$2,000. The prices are further scaled to Market-Sim where agents submit orders priced at integers rather than decimals, and the mean fundamental value is $\bar{r} = 10^5$. The side can change as soon as the imbalance side changes, or for personal reasons of the trader wanting to buy or sell a stock. Once the MSE is scaled prices, the Stock-GAN is not very successful when predicting the price of the next order. Given that I restrict the training set to LOC orders, the next order should be easier to predict, because LOC orders tend to be priced very similarly to the CDA's best available bid and offer, as well as the near indicative price. However, this does not appear to be the case, and the Stock-GAN

likely fails because of the diversity in the dataset. The original study with Stock-GAN by Li et al. (2020) restricted the training set to a single symbol, which reduces the variability in price. This was not possible with the close auction dataset because there is not enough data to train a model and restrict the set to small price range. The Stock-GAN is not very successful in predicting the side of the next order. Thus, it might be difficult for the Stock-GAN to detect patterns in the side feature. It's also possible that excluding other close auction order types made it more difficult for the Stock-GAN to pick up on buy and sell trends, but the stability of the price in most instances made this less of a concern for that feature.

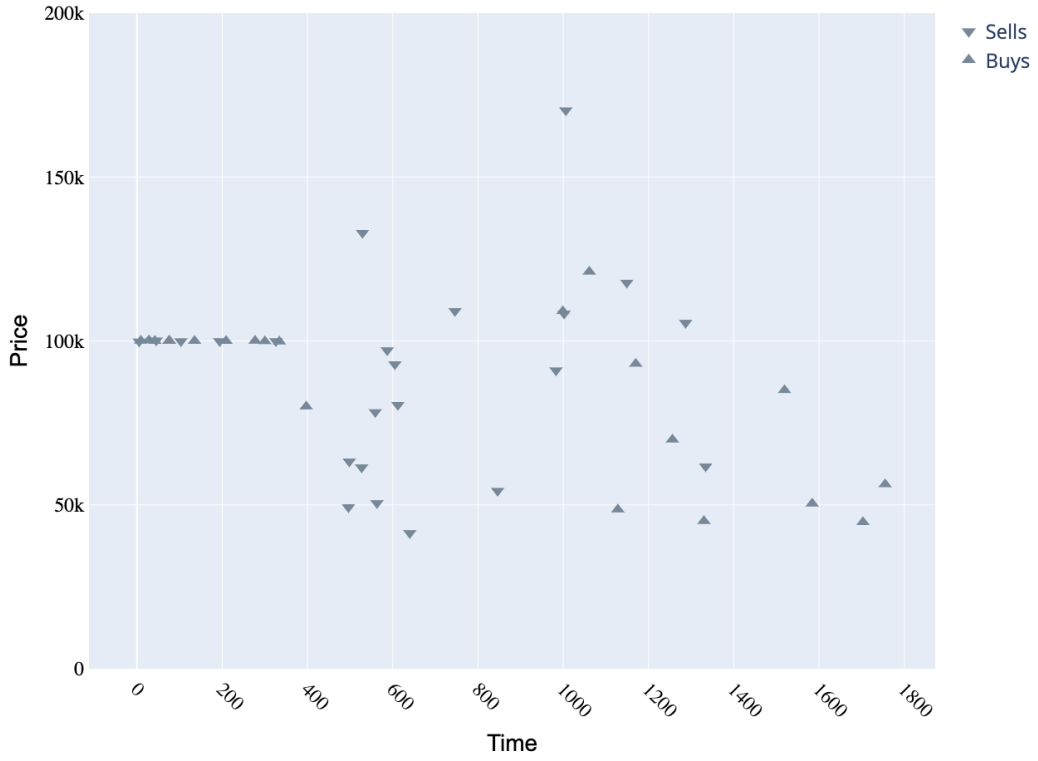


Figure 6.1: A synthetic close order stream generated by Stock-GAN.

Figure 6.1 shows a close auction order stream generated by the Stock-GAN in Market-Sim. A downward arrow represents a sell order, while an upward arrow denotes a buy order. The x-axis is the time step in Market-Sim that a synthetic order is submitted the order to the close auction. The y-axis is the price of the order at the scale of Market-Sim prices. The price and side of the first fifteen orders are randomly generated, and the remaining orders are selected by the Stock-GAN. The side of a

randomly generated order has an equal probability of being a buy or sell order. The price is determined by adding noise from a Gaussian distribution to an estimate of the CDA's final fundamental value. The order stream generated by the Stock-GAN is extremely unrealistic. Most orders are priced very far away from the fundamental value, and the Stock-GAN would have ideally selected prices close to the first fifteen orders in the market priced around the estimated fundamental value. The highest priced buy order is priced significantly higher than the fundamental value, and the lowest priced sell order is priced significantly lower than the fundamental value. GANs are known to be extremely sensitive to model structure and hyperparameter tuning, and extremely difficult to adapt to real data (Quach, 2020). It appears that I was not able to find a setting for the Stock-GAN to efficiently generate close auction orders. This will make it extremely difficult for a benchmark manipulator to influence the close price because it would have to submit orders with prices very far away from their valuation of the asset. Pricing an order extremely far away will likely lead to more of a loss in the market than the manipulator can cover in profits from an external contract dependent on the close price.

6.7.2 Impact of Close Price Manipulation

I test the impact to market welfare when an agent tries to manipulate the close price. I examine a manipulator that uses a standard ZI strategy, a ZIM strategy and learns a strategy through DQN. Table 6.2 shows the various payoffs of agents, the aggregate market payoff, and the close price when the DQN agent deploys these three strategy options. The background surplus is the average payoff of the background traders deploying a ZI strategy. The manipulator total surplus is the combined surplus from the market and close price of the manipulator. A profit from the close price refers to any profit from an external contract whose value is dependent on the close price. The manipulator market surplus refers to its profit solely from the market. The aggregate total surplus is the aggregate market and close price surplus of all trading agents, though the manipulator is the only agent with any external profit dependent on the close in this scenario. The aggregate market surplus is the aggregate market surplus of all trading agents. Lastly, the close price refers to the price the close auction closes at.

Table 6.2 shows that the manipulator's total surplus and the close price decreases when it attempts to manipulate the benchmark. The ZIM agent is the worst performing

Table 6.2: Profit of agents and aggregate market over 200 simulation runs when the DQN agent deploys a ZI strategy versus learning a strategy through DQN.

	ZI	**ZIM**	**DQN**
Background Trader Profit	1,114	**1,159**	1,066
Manipulator Total Profit	**1,199**	-6,351	-2,744
Manipulator Market Profit	**1,199**	829	1,065
Aggregate Total Profit	**13,045**	4,872	7,749
Aggregate Market Profit	**13,045**	12,052	11,558
Close Price	**99,594**	99,415	99,499

agent, which likely occurs because the DQN agent learns not manipulator or submit less aggressive orders than the ZIM agent, because manipulation does not work in the environment. On top of the Stock-GAN's inability to replicate a realistic order stream, it may be possible that a manipulator may have trouble influencing a realistic order stream anyway. There is a significant amount of noise in the real market, and some small changes in the CDA's best available bid and offer and imbalance messages might be considered irrelevant, so the manipulator is not able to shift the close auction order stream. Given these factors of realistic data and the failure of this implementation of the Stock-GAN to generate a reasonable order stream, the manipulator would need to price an order extremely far away from its valuation of the asset. This will likely lead to more of a loss in the market than the manipulator can cover in profits from an external contract dependent on the close price. The manipulator likely could not determine a trading strategy that would shift the close, because there was no distinguishable pattern that would shift the close.

6.8 Conclusion

In this chapter, I expand the work of Chapter V to include a close auction and set the benchmark to the close price. I also train a Stock-GAN on real NASDAQ close auction data to produce a close auction order stream. I find that the Stock-GAN was better able to choose the price of a close auction order than the side. I also find that a manipulator failed to shift the close price and increase its overall profit.

The largest contribution of this work was attempting to incorporate historical data into an agent-based market model. Historical data is extremely complex and rich in information about market activity and interactions. However, it is difficult

to answer the question of what happens when new events or interactions take place. Once any new event or activity enters the market, any historical data after that point is possibly irrelevant if other market participants calibrate their actions to the new activity. Agent-based models address this issue by modeling agents' responses to market activity, but these models tend to extrapolate the problem to the point where it becomes to apply back to a real market. Combining historical data and agent-based model to reap the benefits of both methods will be a useful advance in AI and finance, however solving this problem is not possible within the scope of this work.

One limitation of this study was the loss of information when prepping the dataset. To fit into Market-Sim, I greatly reduced the set of close auction orders by excluding multiple order types and order components. I also did not consider a lot of order types submitted to the CDA which could have greatly changed the outcome of the market. To make the close auction order stream adaptable to Market-Sim, I may have eliminated crucial information to allow the Stock-GAN to learn a realistic order stream. It also makes the results of the problem difficult to apply to real close auction manipulation because there are so many simplifications to the market model and dataset.

Another limitation of this work is the limited testing of the Stock-GAN. GANs are very difficult to train on real data and very sensitive to hyperparameter and model changes (Quach, 2020). I was unable to do a sufficient hyperparameter test on the Stock-GAN because of resource limitations where I could access the data. Therefore, it is highly likely that the Stock-GAN is not as efficient as possible. The original implementation of the Stock-GAN by Li et al. (2020) also only trained the Stock-GAN on a single stock at a time. It was not possible to limit my training set to a signal stock because of a small dataset. Therefore, there was a high variance in the underlying patterns of the stocks studied, which likely made it difficult for the Stock-GAN to learn patterns to mimic when generating synthetic data. It would have also been useful to test other methods for generating synthetic data, such as a variational autoencoder (VAE) (Kingma and Welling, 2013).

Like the Stock-GAN, I was not able to do a hyperparameter search for the DQN agent in this environment. This could have negatively impacted the DQN agent's performance. Overall, the resources available made it difficult to implement this project. For researchers moving forward, it will be important to consider and improve upon their ability to bring complex systems and artificial intelligence and machine learning tools outside of academia to an institution with market data. Researchers

should also consider limiting their projects to working within the resources available at the institution. It is important to continue to collaborate with institutions with this type of market data though because it is the only way to gain access to proprietary and high granularity market data. Institutions that host this type of market data could also benefit from the advanced tools academic researchers use. It is a lucrative relationship which warrants more consideration to address the intersection of AI and finance.

CHAPTER VII

Conclusions

High volatility in stock markets and events such as the Flash Crash on May 6, 2010 have led to a high interest in trading activity around volatility that causes or exacerbates these events. Volatility events like mini flash crashes form profitable opportunities for market participants. Some market participants receive superior market information than others and may receive that information faster than others, allowing these participants to exploit their advantage and respond to market volatility before others are even aware of the event. Other symbols' values are correlated by things like exchange traded funds (ETFs), and some participants can gain from observing changes in one symbol that will impact others. Examining trading practices around mini flash crashes can provide insight into these practices' impact on market welfare.

Previous scandals concerning financial benchmark manipulation, such as the London Interbank Offer Rate (LIBOR) scandal of 2011, has created a push to evaluate the design of benchmarks to be more robust to manipulation attempts. Many market experts and academics have advocated for transaction-based benchmarks, or benchmarks whose value is calculated with market transactions. This possibly leads to another form of manipulation where a market participant party to a contract whose value is dependent on a benchmark may try to influence the value of the benchmark by trading in the market. Using deep reinforcement learning (DRL) to generate trading strategies has also significantly grown in popularity in recent years. Studying whether DRL algorithm designs, like including market profit and profit from a contract dependent on a benchmark in an algorithms reward function, may lead to the algorithm learning a trading strategy that manipulates the benchmark. Determining the impact of different manipulative strategies can give intuition to participants' ability to influence financial

benchmarks.

I analyzed historical data and develop computational models to study trading strategies that attempt to profit from market volatility and manipulate financial benchmarks. I used historical data to analyze trading activity around mini flash crashes. Then I employed two agent-based simulations to model continuous double auctions (CDAs). Within the scope of these market simulators, I modeled various trading strategies to determine their impact on market welfare. I also created a method to deploy trading strategies learned through DRL within a market simulator. Lastly, I combined historical data analysis and an agent-based market model to study market manipulation. This method may help lead to an informative regulatory sandbox to test new market policies before implementation in real markets. The remainder of this chapter provides summaries of the contributions of the four bodies of work presented in this thesis.

The Phases and Catalysts of Mini Flash Crashes In this work, I built a mini flash crash detector to study trade activity during these events. My mini flash crash detection method splits each event into four phases with variable length because none of these events are identical. This dynamic, phase-base approach enabled me to analyze trade activity at the phase transition points. Up to a minute preceding the start of the price drop, I observed that the proportion of intermarket sweep orders gradually increases. My results demonstrated that some participants may insight the price drop, or exercise trend following strategies in anticipation of a price drop. I studied off-exchange subpenny trades as a proxy for retail volume and discover a 278% increase in retail sell volume around the lowest price of the event. My analysis implies that mini flash crashes may adversely affect retail investors. This work provides evidence that some market participants may cause or exacerbate mini flash crashes with ISO sweeps, and that retail investors are potentially harmed by internalization at the lowest price.

Stability Effects of Arbitrage in Exchange Traded Funds This work used an agent-based model to study ETF arbitrage during volatility events like mini flash crashes. I simulated a market with one index-based ETF and two underlying symbols in the ETF's portfolio. The sum of the prices of the underlying symbols represents the market index. I implemented a trading strategy that submits orders if the ETF trading price drifts from the market index. I simulated a mini flash crash by designing

a trader that submits a series of marketable sell orders, similar to an ISO sweep. I used empirical game-theoretic analysis to study this problem when the agents use strategies in an equilibrium setting. I found that when arbitrageurs actively trade, background traders' surplus increases, which demonstrates that other market participants are better off when a trader uses ETF arbitrage trading strategies. Also, when one symbol experiences a mini flash crash and arbitrageurs are active, the other symbol experiences a price change in the opposite direction. Thus, I showed that ETF arbitrage impacts the stability of the ETF's portfolio in my model. My results imply that ETF arbitrage increases volatility in some ways, but decreases it in other ways. I showed that arbitrage does increase the volatility of other symbols in the ETF's portfolio, but it helps the symbol which experienced a mini flash crash recover faster and helps the ETF track the index. Therefore, I demonstrate that ETF arbitrage has both positive and negative impact during volatility events like a mini flash crash.

Benchmark Manipulation This work explored financial benchmark manipulation in a simulated environment. I examined a market with a benchmark manipulator that employs a heuristic strategy that applies a constant price shift to all orders. I also studied two benchmark manipulators that learn trading strategies through deep reinforcement learning. These agents successfully manipulate the market by shifting the benchmark. The payoffs of the benchmark manipulator and other trading agents increased when the manipulator implements its adversarial strategy. Other trading agents benefit because the manipulator is submitting unprofitable orders in the market to increase its profits from the benchmark. The party that loses in this scenario is the one on the other side of the benchmark contract. My findings provided evidence that this type of manipulation deteriorates the quality of the benchmark, making the benchmark a less efficient summary statistic over a market variable. My results show that a trader can learn manipulative trading strategies through DRL without explicitly instructing the algorithm to do so. This suggests that manipulation law would benefit from expanding to include manipulative strategies learned through a RL algorithm, rather than solely focusing on a human's intention to manipulate.

Close Price Manipulation In this work, I used historical data and an agent-based simulation to study close price manipulation. I used an implementation generative adversarial network (GAN), Stock-GAN, to train on historical close auction order data from NASDAQ to generate synthetic close auction order streams. Ultimately, the

dataset of close auction orders was too small to restrict enough to satisfy the sensitivity of GANs, so the synthetic order streams were not what one would expect in a real close auction. I added a close auction with imbalance messages to a market simulator to determine if an agent that learns trading strategies through deep reinforcement learning can manipulate the close price. Given that my implementation of Stock-GAN was not able to produce a realistic order stream, the manipulative agents were not able to learn trading strategies that influenced the close price. In this work I provided valuable insight on combining historical data and agent-based modeling. The combination of these two methods will be a useful advance in AI and finance, however solving this problem was not possible within the scope of this work.

In my dissertation, I used historical data and computational modeling to study the impacts of certain trading strategies on market welfare. I developed a novel method to detect mini flash crashes with historical data and was able to analyze how certain trading practices may exacerbate these events and impact other market participants. The majority of my work utilized agent-based simulation to model various trading strategies and determine these strategies impact on welfare. One the biggest contributions presented in this thesis is the incorporation of trading strategies learned through DRL in a market simulator. I studied the impact design choices within the DRL algorithm that led to the algorithm learning manipulative trading strategies. This provides insight into how market regulators may need to update current manipulation law, which states that a human must intend to manipulate. With increasing advancements in DRL and algorithmic trading, I show evidence that regulators may need to extend manipulation law to include algorithmic design choices that lead to trading strategies that intend to manipulation. Lastly, I attempted to combine historical data and computational modeling to study market manipulation. This provides valuable insight for future work in this area on how to work with sensitive market data in parallel with advanced machine learning and agent-based simulation.

APPENDICES

APPENDIX A

Equilibria in ETF Arbitrage Games

Table A.1: Role-symmetric equilibria in ETF arbitrage games. This table presents a variety of market environments where ETF arbitrageurs are active and inactive. Each row of the table is the probability that an agent in a role plays that strategy. For example, there is a 34.4% chance that a background trader will play strategy $\mathbf{ZIT}_1$ in a **LSHN** environment without active arbitragers. There are 27 background agents and 4 arbitrageurs in this game. Each arbitrageur is in its own role (SA-C, SA-A, MA-C, and MA-A) and deploys one strategy every time depending on if it active or not.

Environment	Active	Background						ETF Arbitrageurs	
		$\mathbf{ZIT}_1$	$\mathbf{ZIT}_2$	$\mathbf{ZIT}_3$	$\mathbf{ZIT}_4$	$\mathbf{ZIT}_5$	$\mathbf{ZIT}_6$	$\varepsilon = 10^3$	$\varepsilon = 10^{12}$
LSHN	Yes	0	0	0	0	0	1.0	1.0	–
LSHN	No	0.344	0	0.656	0	0	0	–	1.0
MSMN	Yes	0	0	0.149	.851	0	0	1.0	–
MSMN	No	0	0.493	0	0.507	0	0	–	1.0
HSLN	Yes	0	0	0	0.569	0.431	0	1.0	–
HSLN	No	0	0	0	0	1.0	0	–	1.0

APPENDIX B

Table of the Benchmark Manipulator's State Space

Table B.1: Description of each state space feature utilized by our market manipulator.

Feature	**Description**
Private bid	The private value of the next unit bought.
Private ask	The private value of the next unit sold.
Market holdings	The agent's current holdings of the traded asset. (If > 0, then bought more units than sold, and if < 0, then sold more units than bought.)
Contract holdings	The agent's holdings from an external contract whose valuation depends on this market. (Multiplied by the direction the agent is more profitable in, i.e. if the agent is better off if the valuation goes up, then this value is positive. If it's better off if the valuation goes down, this value is positive.)
Side	If the agent will submit a buy or sell order (currently all agents flip a coin in market-sim to

Feature	Description
	determine the side).
Final fundamental estimate	An estimate of the final fundamental. (A noisy observation of the mean reverting time series representing the fundamental value of the traded asset.)
Time until end	The number of time steps remaining in the trading period.
Bid omega ratio	Estimates the "favorability" of submitting a buy order at the current time. Ratio of (recent trade prices) higher than (the agent's estimated value of the asset) to (recent trade prices) lower than (the agent's estimated value of the asset). Only considers the last X trades.
Ask omega ratio	Estimates the "favorability" of submitting a sell order at the current time. Ratio of (recent trade prices) higher than (the agent's estimated value of the asset) to (recent trade prices) lower than (the agent's estimated value of the asset). Only considers the last X trades.
Bid size	Depth of book, bid. The number of active buy orders in the market.
Ask size	Depth of book, ask. The number of active sell orders in the market.
Spread	The difference in price between the best available ask and the best available bid in the book. $\min(\text{sell price}) - \max(\text{buy price})$

Feature	Description
Bid vector	An ordered, padded vector of the difference between the price of all active buy orders and the estimated value. Organized in ascending order by price, then time, i.e. the highest-priced bid is the last element, and if two orders have the same price, then the order that arrived first has the higher index. If there are fewer active buy orders than the length of the vector, then it is padded with very low bid prices.
Ask vector	An ordered, padded vector of the difference between the estimated value and the price of all active sell orders. Organized in descending order by price, then time, i.e. the lowest-priced ask is the first element, and if two orders have the same price, then the order that arrived first has the lower index. If there are fewer active sell orders than the length of the vector, then it is padded with very high sell prices.
Number of transactions	The current number of trades that have occurred in the market.
Transaction history	A padded ordered list of the difference between the estimated value and the price trades. Organized in descending order by time, i.e. the most recent trade is the first element. If there are fewer trades than the length of the vector, then it is padded with zeros.

APPENDIX C

Benchmark Manipulation: Deep Reinforcement Learning Hyperparameters

Table C.1: The hyperparameters of DQN when a MM is present.

Hyperparameter	Value
Number of episodes	2,500
Batch size	1,024
Replay capacity	20,000
Minimum replay size	2,500
Number of gradient steps per update	5
Target updated period	30
Polyak update	True
Error clipping	100.0
Size of network	26
Learning rate	1e-6
Exploration schedule	Constant = 0.2
Reward clipping	100
Omega depth	5
Length of bid vector	5
Length of ask vector	5
Length of transaction vector	5

Table C.2: The hyperparameters of DDPG when a MM is present.

Hyperparameter	Value
Number of episodes	70,000
Batch size	512
Replay capacity	20,000
Minimum replay size	10,000
Number of gradient steps per update	10
Target updated period	30
Target network weight	0.005
Discount factor	0.99
Error clipping	1.0
Reward clipping	10,000
Size of network	128
Learning rate	3e-5
Exploration noise	0.1
Action coefficient C	1,050
Benchmark impact χ	500
Omega depth	5
Length of bid vector	5
Length of ask vector	5
Length of transaction vector	5

Table C.3: The hyperparameters of DQN when a MM is **not** present.

Hyperparameter	Value
Number of episodes	2,500
Batch size	1,024
Replay capacity	20,000
Minimum replay size	2,500
Number of gradient steps per update	5
Target updated period	30
Polyak update	False
Error clipping	False
Size of network	26
Learning rate	1e-5
Exploration schedule	Constant = 0.2
Reward clipping	False
Omega depth	5
Length of bid vector	5
Length of ask vector	5
Length of transaction vector	5

Table C.4: The hyperparameters of DDPG when a MM is **not** present.

Hyperparameter	Value
Number of episodes	110,000
Batch size	1,024
Replay capacity	20,000
Minimum replay size	2,500
Number of gradient steps per update	10
Target updated period	30
Target network weight	0.005
Discount factor	0.99
Error clipping	1.0
Reward clipping	40,000
Size of network	64
Learning rate	3e-5
Exploration noise	0.3
Action coefficient C	1,050
Benchmark impact χ	1,250
Omega depth	5
Length of bid vector	5
Length of ask vector	5
Length of transaction vector	5

APPENDIX D

Equilibria and Deviations in Benchmark Manipulation Games

Table D.1: Role-symmetric equilibria in benchmark manipulation games. This table presents a variety of market environments where benchmark manipulators attempt to manipulate or not manipulate. Each row of the table is the probability that an agent in a role plays that strategy. For example, there is a 34.4% chance that a background trader will play strategy $\mathbf{ZI}_1$ in a **LSHN** environment without active arbitragers. There are 10 background agents and 1 manipulator in this game.

Environment	Contract Holdings	Background					Benchmark Manipulator				
		$\mathbf{ZI}_1$	$\mathbf{ZI}_2$	$\mathbf{ZI}_3$	$\mathbf{ZI}_4$	$\mathbf{ZI}_5$	$\mathbf{ZIM}_1$	$\mathbf{ZIM}_2$	$\mathbf{ZIM}_3$	$\mathbf{ZIM}_4$	$\mathbf{ZIM}_5$
LSHN	ψ^-	0	0	1.0	0	0	1.0	–	–	–	–
LSHN	ψ^-	0	0.458	0.542	0	0	–	–	–	1.0	–
LSHN	ψ^+	0	0	1.0	0	0	1.0	–	–	–	–
LSHN	ψ^+	0.124	0.876	0	0	0	–	–	–	1.0	–
MSMN	ψ^-	0	0	0.353	0.112	0.535	1.0	–	–	–	–
MSMN	ψ^-	0.566	0.037	0	0	0.397	–	1.0	–	1.0	–
MSMN	ψ^+	0	0	0.353	0.112	0.535	1.0	–	–	–	–
MSMN	ψ^+	0	0	0	0	1.0	–	–	–	1.0	–
MSMN	ψ^+	0	0.788	0	0.212	0	–	–	–	1.0	–
HSLN	ψ^-	0	0	0	1.0	0	1.0	–	–	–	–
HSLN	ψ^-	0.461	0.473	0.066	0	0	1.0	–	–	–	–
HSLN	ψ^-	0	0.458	0.542	0	0	–	–	–	1.0	–
HSLN	ψ^+	0	0	0	1.0	0	1.0	–	–	–	–
HSLN	ψ^+	0.461	0.473	0.066	0	0	1.0	–	–	–	–
HSLN	ψ^+	0.464	0	0	0.536	0	–	–	–	1.0	–

Table D.2: Environment **B** with a ZIM manipulator. Benchmark manipulator deviation.

Payoff	Benchmark Manipulator							
	ZIM$_1$	**ZIM**$_2$	**ZIM**$_3$	**ZIM**$_4$	**ZIM**$_5$	**ZIM**$_6$	**ZIM**$_7$	**ZIM**$_8$
1084.79	1	0	0	0	0	0	0	0
1274.31	0	1	0	0	0	0	0	0
1288.53	0	0	1	0	0	0	0	0
1191.94	0	0	0	1	0	0	0	0
1106.50	0	0	0	0	1	0	0	0
1228.72	0	0	0	0	0	1	0	0
1310.98	0	0	0	0	0	0	1	0
1188.44	0	0	0	0	0	0	0	1

Table D.3: Environment **B** with a ZIM manipulator. Single player deviation.

Payoff	Background									
	$\mathbf{ZI}_1$	$\mathbf{ZI}_2$	$\mathbf{ZI}_3$	$\mathbf{ZI}_4$	$\mathbf{ZI}_5$	$\mathbf{ZI}_6$	$\mathbf{ZI}_7$	$\mathbf{ZI}_8$	$\mathbf{ZI}_9$	$\mathbf{ZI}_{10}$
1057.98	1	0	0	0	0	0	14	0	0	0
1053.16	0	1	0	0	0	0	14	0	0	0
1061.28	0	0	1	0	0	0	14	0	0	0
1053.04	0	0	0	1	0	0	14	0	0	0
1071.51	0	0	0	0	1	0	14	0	0	0
1061.73	0	0	0	0	0	1	14	0	0	0
1058.02	0	0	0	0	0	0	15	0	0	0
1055.22	0	0	0	0	0	0	14	1	0	0
1054.44	0	0	0	0	0	0	14	0	1	0
1059.99	0	0	0	0	0	0	14	0	0	1

Table D.4: Environment **B** with a ZIM manipulator. Mixed player deviation.

Payoff	Background									
	$\mathbf{ZI}_1$	$\mathbf{ZI}_2$	$\mathbf{ZI}_3$	$\mathbf{ZI}_4$	$\mathbf{ZI}_5$	$\mathbf{ZI}_6$	$\mathbf{ZI}_7$	$\mathbf{ZI}_8$	$\mathbf{ZI}_9$	$\mathbf{ZI}_{10}$
1055.32	0	0	0	0	12	0	3	0	0	0
1046.07	0	0	0	0	10	0	5	0	0	0
1043.70	0	0	0	0	8	0	7	0	0	0
1051.70	0	0	0	0	7	0	8	0	0	0
1049.55	0	0	0	0	5	0	10	0	0	0
154.03	0	0	0	0	3	0	12	0	0	0
1071.51	0	0	0	0	1	0	14	0	0	0

Table D.5: Environment **B** with a ZIM manipulator. Benchmark manipulator deviation.

Payoff	Benchmark Manipulator							
	ZIM$_1$	**ZIM**$_2$	**ZIM**$_3$	**ZIM**$_4$	**ZIM**$_5$	**ZIM**$_6$	**ZIM**$_7$	**ZIM**$_8$
1030.30	1	0	0	0	0	0	0	0
1179.88	0	1	0	0	0	0	0	0
1302.12	0	0	1	0	0	0	0	0
1065.72	0	0	0	1	0	0	0	0
1065.54	0	0	0	0	1	0	0	0
1275.00	0	0	0	0	0	1	0	0
1191.44	0	0	0	0	0	0	1	0
1222.44	0	0	0	0	0	0	0	1

Table D.6: Environment **B** with a ZIM manipulator. Second single player deviation.

Payoff	Background									
	$\mathbf{ZI}_1$	$\mathbf{ZI}_2$	$\mathbf{ZI}_3$	$\mathbf{ZI}_4$	$\mathbf{ZI}_5$	$\mathbf{ZI}_6$	$\mathbf{ZI}_7$	$\mathbf{ZI}_8$	$\mathbf{ZI}_9$	$\mathbf{ZI}_{10}$
1049.47	1	0	0	0	1	0	13	0	0	0
1050.23	0	1	0	0	1	0	13	0	0	0
1055.74	0	0	1	0	1	0	13	0	0	0
1057.71	0	0	0	1	1	0	13	0	0	0
1061.30	0	0	0	0	2	0	13	0	0	0
1066.48	0	0	0	0	1	1	13	0	0	0
1062.47	0	0	0	0	1	0	14	0	0	0
1058.84	0	0	0	0	1	0	13	1	0	0
1053.99	0	0	0	0	1	0	13	0	1	0
1054.64	0	0	0	0	1	0	13	0	0	1

Table D.7: Environment **B** with a ZIM manipulator. Three strategy mixed player deviation.

Payoff	Background									
	$\mathbf{ZI}_1$	$\mathbf{ZI}_2$	$\mathbf{ZI}_3$	$\mathbf{ZI}_4$	$\mathbf{ZI}_5$	$\mathbf{ZI}_6$	$\mathbf{ZI}_7$	$\mathbf{ZI}_8$	$\mathbf{ZI}_9$	$\mathbf{ZI}_{10}$
1048.60	0	0	0	0	5	5	5	0	0	0
1057.09	0	0	0	0	1	4	10	0	0	0
1048.97	0	0	0	0	4	1	10	0	0	0
1052.14	0	0	0	0	1	2	12	0	0	0
1049.40	0	0	0	0	2	1	12	0	0	0
1066.48	0	0	0	0	1	1	13	0	0	0

Table D.8: Environment **B** with a ZIM manipulator. Benchmark manipulator deviation.

Payoff	Benchmark Manipulator							
	ZIM$_1$	**ZIM**$_2$	**ZIM**$_3$	**ZIM**$_4$	**ZIM**$_5$	**ZIM**$_6$	**ZIM**$_7$	**ZIM**$_8$
1096.57	1	0	0	0	0	0	0	0
1235.81	0	1	0	0	0	0	0	0
1330.30	0	0	1	0	0	0	0	0
1212.72	0	0	0	1	0	0	0	0
1120.04	0	0	0	0	1	0	0	0
1117.93	0	0	0	0	0	1	0	0
1225.45	0	0	0	0	0	0	1	0
1132.41	0	0	0	0	0	0	0	1

Table D.9: Environment **B** with a DQN manipulator. Single player deviation.

Payoff	Background									
	$\mathbf{ZI}_1$	$\mathbf{ZI}_2$	$\mathbf{ZI}_3$	$\mathbf{ZI}_4$	$\mathbf{ZI}_5$	$\mathbf{ZI}_6$	$\mathbf{ZI}_7$	$\mathbf{ZI}_8$	$\mathbf{ZI}_9$	$\mathbf{ZI}_{10}$
1055.01	1	0	0	0	0	0	14	0	0	0
1064.48	0	1	0	0	0	0	14	0	0	0
1053.80	0	0	1	0	0	0	14	0	0	0
1053.39	0	0	0	1	0	0	14	0	0	0
1053.09	0	0	0	0	1	0	14	0	0	0
1059.18	0	0	0	0	0	1	14	0	0	0
1053.44	0	0	0	0	0	0	15	0	0	0
1064.49	0	0	0	0	0	0	14	1	0	0
1064.99	0	0	0	0	0	0	14	0	1	0
1051.96	0	0	0	0	0	0	14	0	0	1

Table D.10: Environment **B** with a DQN manipulator. Mixed player deviation.

Payoff	Background									
	$\mathbf{ZI}_1$	$\mathbf{ZI}_2$	$\mathbf{ZI}_3$	$\mathbf{ZI}_4$	$\mathbf{ZI}_5$	$\mathbf{ZI}_6$	$\mathbf{ZI}_7$	$\mathbf{ZI}_8$	$\mathbf{ZI}_9$	$\mathbf{ZI}_{10}$
1058.95	0	0	0	0	0	0	3	0	12	0
1042.39	0	0	0	0	0	0	5	0	10	0
1061.90	0	0	0	0	0	0	7	0	8	0
1054.41	0	0	0	0	0	0	8	0	7	0
1054.43	0	0	0	0	0	0	10	0	5	0
1054.01	0	0	0	0	0	0	12	0	3	0
1064.99	0	0	0	0	0	0	14	0	1	0

APPENDIX E

Close Price Manipulation: Deep Reinforcement Learning Hyperparameters

Table E.1: The hyperparameters of DQN.

Hyperparameter	Value
Number of episodes	2,500
Batch size	1,024
Replay capacity	20,000
Minimum replay size	2,500
Number of gradient steps per update	5
Target updated period	30
Polyak update	True
Error clipping	1.0
Size of network	26
Learning rate	1e-5
Exploration schedule	Constant = 0.2
Reward clipping	False
Omega depth	5
Length of bid vector	5
Length of ask vector	5
Length of transaction vector	5

APPENDIX F

Stock-GAN Model

Model: Memory		
Layer (type)	Output Shape	Param #
lstm_3 (LSTM)	(None, 8)	544
Total params: 544		
Trainable params: 544		
Non-trainable params: 0		

Table F.1: Model summary for Stock-GAN's order history pre-processing.

Model: Generator

Layer (type)	Output Shape	Param #
dense_1 (Dense)	(None, 800)	87200
batch_normalization (BatchNormalization)	(None, 800)	3200
leaky_re_lu_5 (LeakyReLU)	(None, 800)	0
reshape (Reshape)	(None, 1, 8, 100)	0
up_sampling2d (UpSampling2D)	(None, 2, 16, 100)	0
dropout_3 (Dropout)	(None, 2, 16, 100)	0
up_sampling2d_1 (UpSampling2D)	(None, 4, 32, 100)	0
conv2d_transpose (Conv2DTranspose)	(None, 4, 32, 32)	3276832
batch_normalization_1 (BatchNormalization)	(None, 4, 32, 32)	128
leaky_re_lu_6 (LeakyReLU)	(None, 4, 32, 32)	0
conv2d_transpose_1 (Conv2DTranspose)	(None, 4, 32, 16)	524304
batch_normalization_2 (BatchNormalization)	(None, 4, 32, 16)	64
leaky_re_lu_7 (LeakyReLU)	(None, 4, 32, 16)	0
conv2d_transpose_2 (Conv2DTranspose)	(None, 4, 32, 8)	131080
batch_normalization_3 (BatchNormalization)	(None, 4, 32, 8)	32
leaky_re_lu_8 (LeakyReLU)	(None, 4, 32, 8)	0
max_pooling2d (MaxPooling2D)	(None, 2, 16, 8)	0
conv2d_transpose_3 (Conv2DTranspose)	(None, 2, 16, 1)	8193
max_pooling2d_1 (MaxPooling2D)	(None, 1, 8, 1)	0
Total params: 4,031,033		
Trainable params: 4,029,321		
Non-trainable params: 1,712		

Table F.2: Model summary for Stock-GAN's generator.

Model: Discriminator

Layer (type)	Output Shape	Param #
conv2d_2 (Conv2D)	(None, 1, 16, 512)	5120
leaky_re_lu_2 (LeakyReLU)	(None, 1, 16, 512)	0
dropout (Dropout)	(None, 1, 16, 512)	0
conv2d_3 (Conv2D)	(None, 1, 16, 256)	1179904
leaky_re_lu_3 (LeakyReLU)	(None, 1, 16, 256)	0
dropout_1 (Dropout)	(None, 1, 16, 256)	0
conv2d_4 (Conv2D)	(None, 1, 16, 128)	295040
leaky_re_lu_4 (LeakyReLU)	(None, 1, 16, 128)	0
dropout_2 (Dropout)	(None, 1, 16, 128)	0
flatten (Flatten)	(None, 2048)	0
dense (Dense)	(None, 1)	2049
Total params: 1,482,113		
Trainable params: 1,482,113		
Non-trainable params: 0		

Table F.3: Model summary for Stock-GAN's discriminator.

BIBLIOGRAPHY

BIBLIOGRAPHY

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. `https://www.tensorflow.org/` Software available from tensorflow.org.

Carol Alexander and Douglas Cumming. 2020. *Corruption and Fraud in financial markets: Malpractice, Misconduct and Manipulation*. John Wiley & Sons.

Selim Amrouni, Aymeric Moulin, Jared Vann, Svitlana Vyetrenko, Tucker Balch, and Manuela Veloso. 2021. ABIDES-Gym: Gym Environments for Multi-Agent Discrete Event Simulation and Application to Financial Markets. In *3rd ACM International Conference on AI in Finance*.

Stanislav Anatolyev, Sergei Seleznev, and Veronika Selezneva. 2020. Does Index Arbitrage Distort the Market Reaction to Shocks? *SSRN Electronic Journal* (2020), 1–50.

Rochelle Antoniewicz and Jane Heinrichs. 2014. Understanding Exchange-Traded Funds: How ETFs Work. *SSRN Electronic Journal* (2014), 1–40.

Rochelle Antoniewicz and Jane Heinrichs. 2015. The Role and Activities of Authorized Participants of Exchange-Traded Funds. Investment Company Institute.

Matteo Aquilina, Brian Eyles, Jia Shao, and Carla Ysusi. 2018. How do participants

behave during flash events? Evidence from the UK equity market. Financial Conduct Authority.

Stephen Bain, Shary Mudassir, Jennifer Hadiaris, and Michael Liscombe. 2014. The impact of intraday volatility on investor costs. RBC Capital Markets.

Gunjan Banerji. 2018. Regulator Looks into Alleged Manipulation of VIX, Wall Street's 'Fear Index'. *Wall Street Journal* (2018).

Aurelio F. Bariviera, Belén Guercio, Lisana B. Martinez, and Osvaldo A. Rosso. 2016. Libor at crossroads: Stochastic switching detection using information theory quantifiers. *Chaos, Solitons & Fractals* 88 (2016), 172–182.

Robert Battalio and Craig W. Holden. 2001. A simple model of payment for order flow, internalization, and total trading cost. *Journal of Financial Economics* 4, 1 (2001), 33–71.

Erhan Bayraktar and Alexander Munk. 2017. Mini-flash crashes, model risk, and optimal execution. *SSRN Electronic Journal* (2017), 1–48.

Mario Bellia, Kim Christensen, Aleksey Kolokolov, Loriana Pelizzon, and Roberto Renò. 2020. High-Frequency Trading During Flash Crashes: Walk of Fame or Hall of Shame? *SSRN Electronic Journal* (2020), 1–75.

Itzhak Ben-David, Francesco Franzoni, and Rabih Moussawi. 2015. Do ETFs Increase Volatility? Securities and Exchange Commission Comments.

Vineer Bhansali and Lawrence Harris. 2018. Everybody's Doing it: Short Volatility Strategies and Shadow Financial Insurers. *SSRN Electronic Journal* (2018), 1–23.

Elizabeth Blankespoor, Ed deHaan, and Christina Zhu. 2018. Capital market effects of media synthesis and dissemination: evidence from robo-journalism. *Review of Accounting Studies* 23(1) (2018), 1–36.

Ekkehart Boehmer, Charles M. Jones, and Xiaoyan Zhang. 2017. Tracking retail investor activity. *SSRN Electronic Journal* (2017), 1–45.

Philip Bond, Alex Edmans, and Itay Goldstein. 2012. The real effects of financial markets. *Annual Review of Financial Economics* 4, 1 (2012), 339–360.

Graham Bowley. 2010. The Flash Crash, in Miniature. The New York Times.

Travis Box, Ryan Davis, Richard Evans, and Andrew Lynch. 2021. Intraday arbitrage between ETFs and their underlying portfolios. *Journal of Financial Economics* 141, 3 (2021), 1078–1095.

Catherine Boyle. 2014. Forex manipulation: How it worked. CNBC.

Tobias Braun, Jonas A. Fiegen, Daniel C. Wagner, Sebastian M. Krause, and Thomas Guhr. 2018. Impact and recovery process of mini flash crashes: An empirical study. *PLOS One* 13, 5 (2018), e0196920.

Erik Brinkman. 2018. *Understanding financial market behavior through empirical game-theoretic analysis*. Ph.D. Dissertation. University of Michigan.

Jonathan Brogaard, Allen Carrion, Thibaut Moyaert, Ryan Riordan, Andriy Shkilko, and Konstantin Sokolov. 2018. High frequency trading and extreme price movements. *Journal of Financial Economics* 128(2) (2018), 253–265.

David Byrd, Maria Hybinette, and Tucker Hybinette Balch. 2019. ABIDES: Towards High-Fidelity Market Simulation for AI Research. *eprint arXiv:1904.12066* (2019), 1–13.

David Byrd, Maria Hybinette, and Tucker Hybinette Balch. 2020. ABIDES: Towards High-Fidelity Multi-Agent Market Simulation. In *ACM SIGSIM Conference on Principles of Advanced Discrete Simulation* (Miami) *(SIGSIM-PADS '20)*. 11—-22.

Ben Carlson. 2019. Spike, Swoon, Repeat: What to Make of a Volatile Stock Market—That Has Basically Gone Nowhere for a Year and a Half. Fortune.

Ben-Alexander Cassell and Michael P Wellman. 2012. EGTAOnline: An experiment manager for simulation-based game studies. In *International Workshop on Multi-Agent Systems and Agent-Based Simulation*. Springer, 85–100.

CFA Institute. 2012. Dark Pools, Internalization, and Equity Market Quality.

Tanmoy Chakraborty and Michael Kearns. 2011. Market making and mean reversion. In *12th ACM Conference on Electronic Commerce*. 307–314.

Sugato Chakravarty, Pankaj Jain, James Upson, and Robert Wood. 2012. Clean sweep: Informed trading through intermarket sweep orders. *Journal of Financial and Quantitative Analysis* 47(2) (2012), 415–435.

Kim Christensen, Roel Oomen, and Roberto Renò. 2017. The drift burst hypothesis. *SSRN Electronic Journal* (2017), 1–58.

Carole Comerton-Fordea and Tālis J. Putninš. 2011. Measuring closing price manipulation. *Journal of Financial Intermediation* 20, 2 (2011), 135–158.

Zhi Da and Sophie Shive. 2017. Exchange traded funds and asset return correlations. *European Financial Management* 24, 1 (2017), 136–168.

Yue Deng, Feng Bao, Youyong Kong, Zhiquan Ren, and Qionghai Dai. 2017. Deep Direct Reinforcement Learning for Financial Signal Representation and Trading. *IEEE Transactions on Neural Networks and Learning Systems* 28, 3 (2017), 653–664.

Darrell Duffie. 2018. *Compression Auctions with an Application to LIBOR-SOFR Swap Conversion.* Working Paper 3727. Stanford Graduate School of Business.

Darrell Duffie and Piotr Dworczak. 2018. *Robust Benchmark Design.* Working Paper 3175. Stanford Graduate School of Business.

Darrell Duffie and Jeremy C. Stein. 2015. Reforming LIBOR and Other Financial Benchmarks. *Journal of Economic Perspectives* 29, 2 (2015), 191–212.

Kate Duguid and Eric Platt. 2022. Volatility in US Treasuries hits highest point since March 2020. Financial Times.

Alexander Eisl, Rainer Jankowitsch, and Marti G. Subrahmanyam. 2017. The Manipulation Potential of Libor and Euribor. *European Financial Management* 23 (2017), 604–647.

Katrina Ellis, Roni Michaely, and Maureen O'Hara. 2000. The accuracy of trade classification rules: Evidence from NASDAQ. *Journal of Financial and Quantitative Analysis* 35, 4 (2000), 529–551.

Robert Engle and Debojyoti Sarkar. 2006. Premiums-Discounts and Exchange Traded Funds. *Journal of Derivatives* 13, 4 (2006), 27–45.

J. Doyne Farmer, Paolo Patelli, and Ilija I. Zovko. 2005. The predictive power of zero intelligence in financial markets. *National Academy of Sciences* 102, 6, 2254–2259.

Maureen Farrell. 2013. Mini flash crashes: A dozen a day. CNN Money.

Charles Favreau and Ryan Garvey. 2019. An Examination of Cross-Market Arbitrage. *Journal of Applied Business and Economics* 21, 3 (2019).

Christoph Frei and Joshua Mitra. 2021. Optimal closing benchmarks. *Finance Research Letters* 40 (2021), 101674.

Gary L. Gastineau. 2004. The Benchmark Index ETF Performance Problem. *Journal of Portfolio Management* 30, 2 (2004), 96–103.

Tyler Gellasch and Chris Nagy. 2019. Benchmark-Linked Investments: Managing Risks and Conflicts of Interest. Healthy Markets Association.

Dhananjay K. Gode and Shyam Sunder. 1993. Allocative Efficiency of Markets with Zero-Intelligence Traders: Market as a Partial Substitute for Individual Rationality. *Journal of Political Economy* 101, 1 (1993), 119–137.

Anton Golub, John Keane, and Ser-Huang Poon. 2017. High frequency trading and mini flash crashes. *SSRN Electronic Journal* (2017), 1–22.

Ben Golub, Barbara Novick, Ananth Madhavan, Ira Shapiro, Kristen Walters, and Maurizio Ferconi. 2013. Exchange Traded Products: Overview, benefits, and myths. Black Rock.

Jorge Goncalves, Roman Kräussl, and Vladimir Levin. 2019. Do 'Speed Bumps' Prevent Accidents in Financial Markets? *SSRN Electronic Journal* (2019), 1–29.

John M. Griffin and Amin Shams. 2018. Manipulation in the VIX? *Review of Financial Studies* 31 (2018), 1377–1417.

Thomas Heath. 2018. The warning from JP Morgan about flash crashes ahead. The Washington Post.

Derek Horstmeyer. 2020. The perfect storm: Why the coronavirus market is particularly dangerous for individual investors. Fortune.

Tsung-Yu Hsieh. 2015. Information disclosure and price manipulation during the pre-closing session: evidence from an order-driven market. *Frontiers in Finance and Economics* 47, 43 (2015), 4670–4684.

ICE Benchmark Administration Limited. 2019. U.S. Dollar ICE Bank Yield Index. Intercontinental Exchange.

IOSCO. 2013. Principles for Financial Benchmarks. The Board of the International Organization of Securities Commissions.

Neil Johnson, Guannan Zhao, Eric Hunsader, Hong Qi, Nicholas Johnson, Jing Meng, and Brian Tivnan. 2013. Abrupt rise of new machine ecology beyond human response time. *Scientific Reports* 2627 (2013), 1–11.

Talia Kaplan. 2022. Volatility in markets will stick around for 'a long time': Investment expert. Fox Business.

Tae Kim. 2018. Goldman Sachs says computerized trading may make the next 'Flash Crash' worse. CNBC.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114* (2013).

Guray Kucukkocaoglu. 2008. Intra-Day Stock Returns and Close-End Price Manipulation In the Istanbul Stock Exchange. *Frontiers in Finance and Economics* 5, 1 (2008), 46–84.

Floris Laly and Mikael Petitjean. 2020. Mini flash crashes: Review, taxonomy and policy responses. *Bulletin of Economic Research* 72(3) (2020), 251–271.

Sandrine Jacob Leal and Mauro Napoletano. 2019. Market stability vs. market resilience: Regulatory policies experiments in an agent-based model with low- and high-frequency trading. *Journal of Economic Behavior and Organization* 157 (2019), 15–41.

Blake LeBaron, Brian Arthur, and Richard Palmer. 1999. Time series properties of an artificial stock market. *Journal of Economic Dynamics and Control* 23, 9–10 (1999), 1487–1516.

Matt Levine. 2018. Carl Icahn Wants to Fight Dell Again: Also Robinhood, Uber, crypto custody and pre-dating. Bloomberg.

Junyi Li, Xintong Wang, Yaoyang Lin, Arunesh Sinha, and Michael P. Wellman. 2020. Generating realistic stock market order streams. In *34th AAAI Conference on Artificial Intelligence*. 727–734.

Yun Li. 2019. Greenspan says the stock market will determine whether there's a recession. CNBC News.

Yang Li, Wanshan Zheng, and Zibin Zheng. 2019. Deep Robust Reinforcement Learning for Practical Algorithmic Trading. *IEEE Access* 7 (2019), 108014–108022.

Timothy Lillicrap, Jonathan Hunt, Alexander Pritzel, Nicolas Heess, Tom Erez, Yuval Tassa, David Silver, and Daan Wierstra. 2016. Continuous control with deep reinforcement learning. In *4th International Conference on Learning Representations*.

Yang Liu, Qi Liu, Hongke Zhao, Zhen Pan, and Chuanren Liu. 2020. Adaptive Quantitative Trading: An Imitative Deep Reinforcement Learning Approach. *AAAI Conference on Artificial Intelligence* 34, 02, 2128–2135.

Hailey Lynch, Sébastien Page, Robert A. Panariello, James A. Tzitzouris Jr., and David Giroux. 2019. The Revenge of the Stock Pickers. *Financial Analysts Journal* 75, 2 (2019), 34–43.

Ananth Madhavan. 2012. Exchange-traded funds, market structure, and the Flash Crash. *Financial Analysts Journal* 68(4) (2012), 20–35.

Ananth Madhavan and Daniel Morillo. 2018. The Impact of Flows into Exchange-Traded Funds: Volumes and Correlations. *Journal of Portfolio Management* 44, 7 (2018), 96–107.

Ben Marshall, Nhut H. Nguyen, and Nuttawat Visaltanachoti. 2013. ETF arbitrage: Intraday evidence. *Journal of Banking and Finance* 37, 9 (2013), 3486–3498.

Annie Massa. 2018. Trader VIP Clubs, 'Ping Pools' Take Dark Trades to New Level. Bloomberg.

James McBride. 2016. Understanding the Libor Scandal. Council on Foreign Relations.

Takanobu Mizuta. 2019. Agent-Based Model of Liquidity and Arbitrage Cost Between ETF and Stocks. In *2019 8th International Congress on Advanced Applied Informatics (IIAI-AAI)*. 685–688.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A. Rusu, Joel Veness, Marc G. Bellemare, Alex Graves, Martin Riedmiller, Andreas K. Fidjeland, Georg Ostrovski, Stig Petersen, Charles Beattie, Amir Sadik, Ioannis Antonoglou, Helen

King, Dharshan Kumaran, Daan Wierstra, Shane Legg, and Demis Hassabis. 2015. Human-level control through deep reinforcement learning. *Nature* 518 (2015), 529–533.

John Moody, Lizhong Wu, Yuansong Liao, and Matthew Saffell. 1998. Performance functions and reinforcement learning for trading systems and portfolios. *Journal of Forecasting* 17, 5–6 (1998), 441–470.

Abhishek Nan, Anandh Perumal, and Osmar R. Zaiane. 2020. Sentiment and Knowledge Based Algorithmic Trading with Deep Reinforcement Learning. arXiv:2001.09403 [cs.AI]

Nanex. 2011. Flash equity failures in 2006, 2007, 2008, 2009, 2010, and 2011.

Nanex Research. 2012. Sub-penny Price Anomaly.

Yuriy Nevmyvaka, Yi Feng, and Michael Kearns. 2006. Reinforcement learning for optimized trade execution. In *23rd International Conference on Machine learning*. 673—-680.

Barbara Novick, Ananth Madhavan, Samara Cohen, Sal Samandar, Sander Van Nugteren, and Alexis Rosenblum. 2017. A Primer on ETF Primary Trading and the Role of Authorized Participants. BlackRock.

Mark Paddrik, Roy Hayes, Andrew Todd, Steve Yang, Peter Beling, and William Scherer. 2012. An agent based model of the E-Mini S P 500 applied to flash crash analysis. In *2012 IEEE Conference on Computational Intelligence for Financial Engineering Economics (CIFEr)*. 1–8.

Marco Pagano, Antonio Sánchez Serrano, and Josef Zechner. 2019. Can ETFs contribute to systemic risk? *Reports of the Advisory Scientific Committee* 9 (2019), 1–40.

Christine A. Parlour and Uday Rajan. 2003. Payment for order flow. *Journal of Financial Economics* 68(3) (2003), 379–411.

James Paulin, Anisoara Calinescu, and Michael Wooldridge. 2018. Agent-Based Modeling for Complex Financial Systems. *IEEE Intelligent Systems* 33, 2 (2018), 74–82.

Eric Platt and Joe Rennison. 2022. Traders struggle to transact shares in volatile US stock market. Financial Times.

E. S. Ponomarev, I. V. Oseledets, and A. S. Cichocki. 2019. Using Reinforcement Learning in the Algorithmic Trading Problem. *Journal of Communications Technology and Electronics* 64, 12 (2019), 1450–1457.

James Poterba and John Shoven. 2002. Exchange Traded Funds: A new investment option for taxable investors. *American Economic Review* 92, 2 (2002), 422–427.

Katyanna Quach. 2020. Good luck using generative adversarial networks in real life – they're difficult to train and finicky to fix. The Register.

Bernhard Rauch, Max Goettsche, and Florian El Mouaaouy. 2013. LIBOR Manipulation: Empirical Analysis of Financial Market Benchmarks using Benford's Law. *SSRN Electronic Journal* (2013).

G. A. Rummery and Mahesan Niranjan. 1994. *On-Line Q-Learning Using Connectionist Systems*. Technical Report.

Jo Saakvitne. 2016. 'Banging the Close': Price Manipulation or Optimal Execution? *SSRN Electronic Journal* (2016), 1–43.

L. Julian Schvartzman and Michael P. Wellman. 2009. Stronger CDA Strategies through Empirical Game-Theoretic Analysis and Reinforcement Learning. In *8th International Conference on Autonomous Agents and Multiagent Systems*.

Megan Shearer. 2020. The Phases and Catalysts of Mini Flash Crashes. *SSRN Electronic Journal* (2020), 1–44.

Megan Shearer, David Byrd, Tucker H Balch, and Michael P Wellman. 2021. Stability Effects of Arbitrage in Exchange Traded Funds: An Agent-Based Model. In *2nd ACM International Conference on Artificial Intelligence in Finance*.

Alexander A. Sherstov and Peter Stone. 2004. Three automated stock-trading agents: A comparative study. In *AAMAS-04 Workshop on Agent-Mediated Electronic Commerce*. 173–187.

Chyng Wen Tee and Christopher Hian Ann Ting. 2020. Cross-Section of Mini Flash Crashes and Their Detection by a State-Space Approach. *SSRN Electronic Journal* (2020), 1–43.

Thibaut Théate and Damien Ernst. 2020. An Application of Deep Reinforcement Learning to Algorithmic Trading. arXiv:2004.06627 [q-fin.TR]

Takuma Torii, Kiyoshi Izumi, and Kenta Yamada. 2016. Shock transfer by arbitrage trading: Analysis using multi-asset artificial market. *Evolutionary and Institutional Economics Review* 12, 2 (2016), 395–412.

Jonathan Tse, Xiang Lin, and Drew Vincent. 2018. AES Analysis: High Frequency Trading—Measurement, Detection and Response.

U.S. Securities and Exchange Commission. 1934. The Securities Exchange Act of 1934 Section 9(a)(2).

U.S. Securities and Exchange Commission. 2005. Regulation National Market System Rule600(b)(30).

U.S. Securities and Exchange Commission. 2010. Findings regarding the market events of May 6, 2010.

U.S. Securities and Exchange Commission. 2018b. Notice of Filing of the Eighteenth Amendment to the National Market System Plan to Address Extraordinary Market Volatility by Cboe BYX Exchange, Inc., Cboe BZX Exchange, Inc., Cboe EDGA Exchange, Inc., Cboe EDGX Exchange, Inc., Chicago Stock Exchange, Inc., Financial Industry Regulatory Authority, Inc., Investors Exchange LLC, NASDAQ BX, Inc., NASDAQ PHLX LLC, The Nasdaq Stock Market LLC, NYSE National, Inc., New York Stock Exchange LLC, NYSE American LLC, and NYSE Arca, Inc. (Release No. 34-84843; File No. 4-631).

Andrew Verstein. 2015. Benchmark Manipulation. *Boston College Law Review* 56 (2015), 215–272.

Iryna Veryzhenko and Nathalie Oriol. 2012. Post Flash Crash Recovery : An Agent-based Analysis. In *Proceedings for the 8th International Conference on Agents and Artificial*, Vol. 17. 190–197.

Tommi A. Vuorenmaa and Liang Wang. 2014. An Agent-Based Model of the Flash Crash of May 6, 2010, with Policy Implications. *SSRN Electronic Journal* (2014), 1–39.

Elaine Wah, Stan Feldman, Francis Chung, Allison Bishop, and Daniel Aisen. 2019. *A comparison of execution quality across U.S. stock exchanges*. Global Algorithmic Capital Markets, Chapter 5, 91–146.

Elaine Wah and Michael P Wellman. 2013. Latency arbitrage, market fragmentation, and efficiency: A two-market model. In *14th ACM conference on Electronic commerce*. 855–872.

Elaine Wah, Mason Wright, and Michael P Wellman. 2017. Welfare effects of market making in continuous double auctions. *Journal of Artificial Intelligence Research* 59 (2017), 613–650.

Xintong Wang, Yevgeniy Vorobeychik, and Michael P. Wellman. 2018. A Cloaking Mechanism to Mitigate Market Manipulation. In *27th International Joint Conference on Artificial Intelligence*. 541–547.

Xintong Wang and Michael P Wellman. 2017. Spoofing the limit order book: An agent-based model. In *16th Conference on Autonomous Agents and MultiAgent Systems*. 651–659.

Xintong Wang and Michael P Wellman. 2020. Market manipulation: An adversarial learning framework for detection and evasion. In *29th International Joint Conference on Artificial Intelligence*.

Michael P. Wellman. 2016. Putting the agent in agent-based modeling. *Autonomous Agents and Multi-Agent Systems* 30 (2016), 1175–1189.

Michael P. Wellman and Elaine Wah. 2017. Strategic agent-based modeling of financial markets. *The Russel Sage Foundation Journal of Social Sciences* 3, 1 (2017), 104–119.

Bryce Wiedenbeck and Michael P. Wellman. 2012. Scaling simulation-based game analysis through deviation-preserving reduction. In *11th International Conference on Autonomous Agents and Multiagent Systems*. 931–938.

Mason Wright and Michael P. Wellman. 2018. Evaluating the Stability of Non-Adaptive Trading in Continuous Double Auctions. In *17th International Conference on Autonomous Agents and Multiagent Systems*.

Xing Wu, Haolei Chen, Jianjia Wang, Luigi Troiano, Vincenzo Loia, and Hamido Fujita. 2020. Adaptive stock trading strategies with deep reinforcement learning methods. *Information Sciences* 538 (2020), 142–158.

Michael Wursthorn, Mischa Frankl-Duval, and Gregory Zuckerman. 2020. Everyone's a Day Trader Now. The Wall Street Journal.

Zhuoran Xiong, Xiao-Yang Liu, Shan Zhong, Hongyang Yang, and Anwar Walid. 2018. Practical Deep Reinforcement Learning Approach for Stock Trading. arXiv:1811.07522 [cs.LG]

Hongyang Yang, Xiao-Yang Liu, Shan Zhong, and Anwar Walid. 2020. Deep Reinforcement Learning for Automated Stock Trading: An Ensemble Strategy. *SSRN Electronic Journal* (2020), 1–9.

Linlin Ye. 2016. Understanding the Impacts of Dark Pools on Price Discovery. *SSRN Electronic Journal* (2016), 1–67.

Zihao Zhang, Stefan Zohren, and Stephen Roberts. 2020. Deep Reinforcement Learning for Trading. *The Journal of Financial Data Science* 2, 2 (2020), 25–40.