

MICROSTRUCTURE AND MACROBEHAVIOR IN THE MODERN U.S. EQUITY MARKET

A Dissertation Presented

by

Ethan Ratliff-Crain

to

The Faculty of the Graduate College

of

The University of Vermont

In Partial Fulfillment of the Requirements
for the Degree of Doctor of Philosophy
Specializing in Complex Systems and Data Science

August, 2026

Defense Date: July 8, 2026
Dissertation Examination Committee:

Brian Tivnan, Ph.D., Advisor
Christopher Danforth, Ph.D., Co-Advisor
David Novak, Ph.D., Chairperson
Peter Dodds, Ph.D.
Laurent Hébert-Dufresne, Ph.D.
Christopher L. Berger, Ph.D., Dean of The Graduate College

The author's affiliation with The MITRE Corporation is provided for identification purposes only and is not intended to convey or imply MITRE's concurrence with, or support for, the positions, opinions, or viewpoints expressed by the author.

©2026 The MITRE Corporation. ALL RIGHTS RESERVED.
Approved for Public Release; Distribution Unlimited. Public
Release Case Number 26-1631.

ABSTRACT

The U.S. equity market, also known as the National Market System (NMS), has undergone significant change in recent decades as a result of technological, regulatory, and structural shifts. These changes to the underlying organization of the market raise the questions of whether our prior understanding of market behavior still holds and how modern market structure affects aggregate dynamics. Uncovering how changes at lower levels of a system’s architecture affect higher-level behavior is crucial for improving our understanding of the system, informing interventions, and predicting outcomes in real-world scenarios. This dissertation investigates aggregate price dynamics in the context of the NMS and seeks to relate these dynamics back to features of its underlying structure.

We first test whether the statistical properties of price changes (returns) as documented in the latter half of the twentieth century still hold for stock returns observed in the modern NMS. Despite the complex process by which markets establish an evolving price for an asset, empirical regularities of this process have been documented across many markets and time periods. We revisit a seminal study from 2001 which summarized 11 recurring ‘stylized facts’ of returns. We test all 11 of these properties for intraday returns measured on the individual stocks in the Dow 30, using 101 days of transaction data from October 2018–March 2019. We find evidence in support of eight of these stylized facts and no support for the other three. To our knowledge, this also represents the first test of these 11 stylized facts against a consistent set of stocks. Together, our results clarify how these properties should be interpreted in the modern NMS.

We next examine the effects of a specific feature of the modern NMS: the fragmentation of trading across many venues. We independently replicate a leading agent-based model of latency arbitrage in a fragmented market. Through this replication, we successfully reproduce most of the qualitative behavior from the original model while uncovering nuances to its results. We then extend the model to examine whether and how structural changes introduced by market fragmentation drive changes in aggregate price dynamics relative to a consolidated-market baseline. In both of these computational investigations, we document how changes to the information perceived and acted on by traders can materially affect aggregate outcomes in a market. Finally, we conclude by discussing how this cumulative approach of replication and extension can, over time, advance our understanding of market behavior and its relation to underlying structure.

CITATIONS

Material from this dissertation has been published in the following forms:

Ratliff-Crain, E., Van Oort, C. M., Koehler, M. T. K., & Tivnan, B. F.. (2025). Revisiting Cont's stylized facts for modern stock markets. *Quantitative Finance*, 25(9), 1343–1373. <https://doi.org/10.1080/14697688.2025.2530637>

Material from this dissertation has been submitted in the following forms:

Ratliff-Crain, E., Van Oort, C. M., Koehler, M. T. K., & Tivnan, B. F.. (2026). Testing replication for an agent-based model of market fragmentation and latency arbitrage. Submitted to journal.

Ratliff-Crain, E., Van Oort, C. M., Koehler, M. T. K., & Tivnan, B. F.. (2026). How does market fragmentation affect price dynamics? Evidence from an agent-based model. Submitted to journal.

Completing a PhD involves spending a lot of time in one's own head. Thank you to Nicole, Briggsby, and Howl for letting me know when it was time to get out of mine.

ACKNOWLEDGEMENTS

First, I am very thankful to my advisors and committee for their mentorship, support, and for helping shape my research perspective: Brian Tivnan, Chris Danforth, Peter Dodds, Laurent Hébert-Dufresne, and David Novak. Thank you as well to the broader Vermont Complex Systems Institute (VCSI) for providing such a vibrant place full of great people with which to learn and explore. Thank you to MITRE for supporting my research and providing me with excellent mentors, coauthors, and colleagues: Colin Van Oort, Matt Koehler, Matt McMahon, Matt Mihalcin, Zoe Henscheid, and many more. Finally, thank you to my family for their support over the years, and thank you to Cornelia Osborne for encouraging me to pursue this.

TABLE OF CONTENTS

Citations	ii
Dedication	iii
Acknowledgements	iv
List of Figures	xiii
List of Tables	xvi
1 Introduction	1
1.1 Background	3
1.2 Aggregate price dynamics	5
1.2.1 Generative models of aggregate behavior	7
1.2.2 Synthesis	10
2 Revisiting Cont's Stylized Facts for Modern Stock Markets	12
Abstract	12
2.1 Introduction	13
2.1.1 Background	14
2.1.2 Cont's Stylized Facts	16
2.1.3 The Modern U.S. Stock Market	23
2.2 Methodology	26
2.3 Results	30
2.3.1 Linear Autocorrelation of Returns	31
2.3.2 Heavy Tails and Aggregational Gaussianity	32
2.3.3 Gain/Loss Asymmetry	42
2.3.4 Volatility Clustering	43
2.3.5 Intermittency	45
2.3.6 Leverage Effect	47
2.3.7 Volume/Volatility Correlation	49
2.3.8 Asymmetry in Timescales	51
2.4 Discussion	53
3 Replicating an ABM of Market Fragmentation and Latency Arbitrage	56
Abstract	56
3.1 Introduction	57
3.2 Background	60
3.2.1 Replication of agent-based market models	61
3.3 The model	63

3.3.1	Entities, state variables, and scales	65
3.3.2	Process overview and scheduling	68
3.3.3	Exchanges	68
3.3.4	Security Information Processor (SIP)	71
3.3.5	Security	71
3.3.6	Latency arbitrageur (LA) trader	72
3.3.7	Zero-intelligence (ZI) traders	73
3.3.8	Model experiments	77
3.3.9	Output metrics	80
3.4	Quantitative alignment	83
3.4.1	Methodology	84
3.4.2	<i>BestGuess</i> versus WW	88
3.4.3	Using an existing codebase: <i>MarketSim</i>	90
3.4.4	Differences between <i>MarketSim</i> and our <i>BestGuess</i> implemen- tation	96
3.4.5	<i>BestGuess+MarketSim</i>	100
3.5	Qualitative behavior of the model	104
3.5.1	Surplus	105
3.5.2	Execution time	107
3.5.3	Spread	110
3.5.4	Transactions	113
3.5.5	Summary of results	115
3.6	Conclusion	117
4	How Does Market Fragmentation Affect Price Dynamics?	120
	Abstract	120
4.1	Introduction	121
4.2	Background	125
4.2.1	Financial ABMs	129
4.3	Model overview	134
4.3.1	Information regimes	139
4.4	Alignment and extension	140
4.4.1	WW+GetReal	143
4.4.2	WW+GetReal experiments	150
4.4.3	Prices, returns, and volumes	151
4.4.4	Verification of WW+GetReal integration	152
4.5	Fragmentation experiment results	163
4.5.1	Price deviation from consolidated baseline	170
4.5.2	Fragmentation and the statistical properties of returns	173
4.6	Discussion	186

4.6.1	Limitations	188
4.6.2	Future work	191
5	Conclusion	193
	Bibliography	196
A	Supplementary Material for Chapter 2	214
A.1	Results on Raw Versus Normalized Returns	214
A.2	Theoretical Versus Bootstrap Confidence Bands	218
A.3	Autocorrelation of Trading Volume	219
A.4	Volume/Volatility Correlation — Additional Details	220
A.5	Supplementary Results	221
A.6	Alternative visualizations of scaling relationships	225
A.7	Asymmetry in timescales — alternative measures	229
A.8	Interarrival times	231
A.9	Tail fits and distributional comparisons	234
B	Supplementary Material for Chapter 3	241
B.1	Using <i>MarketSim</i> to replicate WWW	241
B.2	Testing model implementation with order routing bug	246
B.3	ODD Protocol	254
B.3.1	Overview	255
B.3.2	Design Concepts	257
B.3.3	Initialization	260
B.3.4	Input data	262
B.3.5	Parameters	262
B.3.6	Simulation Driver	266
B.3.7	Scheduler	266
B.3.8	Exchanges	267
B.3.9	The SIP	269
B.3.10	Security	270
B.3.11	Zero Intelligence (ZI)	271
B.3.12	Latency Arbitrage (LA)	276
B.3.13	Metric calculation	277
C	Supplementary Material for Chapter 4	280
C.1	ODD Protocol	280
C.1.1	Overview	281
C.1.2	Design Concepts	285
C.1.3	Initialization	288

C.1.4	Input data	289
C.1.5	Parameters	290
C.1.6	Simulation Driver	293
C.1.7	Scheduler	293
C.1.8	Exchanges	294
C.1.9	The SIP	297
C.1.10	Observer	299
C.1.11	Security	300
C.1.12	Zero Intelligence (ZI)	302
C.1.13	Latency Arbitrageur (LA)	306
C.1.14	Noise agents	307
C.1.15	Value agent	310
C.1.16	Market Maker Agent	313
C.1.17	Momentum Agent	315
C.1.18	Metric calculation	317
C.2	Replication and Validation of Wah and Wellman (2016)	319
C.2.1	Wah and Wellman’s 2016 model	320
C.2.2	WW model experiments	325
C.2.3	Verification of WW	327
C.2.4	Validation of WW	329
C.3	<i>base</i> and <i>diverse</i> fragmentation results	337
C.3.1	RAD results	338

LIST OF FIGURES

1.1	Example intraday price series and the corresponding log-returns. . . .	4
2.1	The U.S. National Market System (NMS) circa 2018.	26
2.2	Linear autocorrelation of the normalized returns.	32
2.3	Kurtosis as a function of timescale for the return distributions. . . .	36
2.4	Power-law tail exponent values estimated for the return distribution tails.	39
2.5	Stacked bar chart comparing distributional fits for return distribution tails across timescales.	41
2.6	Skew of the normalized return distributions in clock- and event-time.	43
2.7	Autocorrelation of absolute normalized returns.	44
2.8	Leverage effect $L(\tau, \Delta t) = \text{corr}(r(t, \Delta t), r(t + \tau, \Delta t))$ for the nor- malized returns.	48
2.9	Volume versus lagged volatility: $V(\tau, \Delta t) = \text{corr}(\text{volume}(t, \Delta(t)), r(t + \tau, \Delta t))$	49
2.10	Average fine versus coarse volatility, $A(\tau, \Delta t, \Delta T) = \text{corr}(r(t \in T, \Delta t) , r(T + \tau, \Delta T))$	52
2.11	Asymmetry effect—The difference between positive and negative lags for the correlation between the average fine volatility and lagged coarse volatility.	53
3.1	Visualization of WW model following the visual ODD (vODD) format.	66
3.2	High-level logic for a simple continuous double auction (CDA) market.	70
3.3	Mean surplus results from our implementations of the model and <i>Mar-</i> <i>ketSim</i> versus the figures reported by WW.	106
3.4	Mean execution time results from our implementations of the model and <i>MarketSim</i> versus the figures reported by WW.	108
3.5	Mean median spreads from the exchange BBOs for our implementa- tions of the model and <i>MarketSim</i> versus the figures reported by WW.	110
3.6	Mean median SIP NBBO spread for our implementations of the model and <i>MarketSim</i> versus the figures reported by WW.	112
3.7	Mean number of transactions from our implementations of the model and <i>MarketSim</i> versus the figures reported by WW.	114
4.1	High-level overview of the modeled market structure with two ex- changes and one latency arbitrageur (LA).	134
4.2	Example comparing a scenario of orders placed in a consolidated versus fragmented market.	137

4.3	Histograms of the one-minute and ten-minute return distributions from the <i>base</i> and <i>diverse</i> agent configurations of WW+GetReal.	154
4.4	Local autocorrelation at a single lag for a rolling 30-minute window, calculated on 1-minute returns for the <i>base</i> and <i>diverse</i> agent configurations.	155
4.5	Average of the sample-level squared-return autocorrelation results for simulations of the <i>base</i> and <i>diverse</i> agent configuration of WW+GetReal.	156
4.6	Contemporaneous volume/volatility correlation results for the <i>base</i> and <i>diverse</i> agent configuration of WW+GetReal.	157
4.7	Histogram of the number of quotes in 1-minute time windows, with gamma distribution fits, for the <i>base</i> and <i>diverse</i> agent configuration of WW+GetReal.	159
4.8	Histograms of the distribution of quote interarrival times for the <i>base</i> and <i>diverse</i> agent configurations of WW+GetReal.	160
4.9	Normalized intraday quote activity level results with quadratic fits for the <i>base</i> and <i>diverse</i> agent configuration of WW+GetReal.	161
4.10	Paired price paths for the <i>base2x</i> agent configuration under the NBBO information regime with a common random seed (8170772).	166
4.11	Paired price paths for the <i>base2x</i> agent configuration under the BBO information regime with a common random seed (8170772).	167
4.12	Paired price paths for the <i>diverse2x</i> agent configuration under the NBBO information regime with a common random seed (8170772).	168
4.13	Paired price paths for the <i>diverse2x</i> agent configuration under the BBO information regime with a common random seed (8170772).	169
4.14	Relative absolute deviation (RAD) versus consolidated-market baseline without latency.	172
4.15	CCDFs of the absolute return distributions generated by market configurations where the trader agents use the SIP NBBO for their market information.	177
4.16	CCDFs of the absolute return distributions generated by market configurations where the trader agents use their local exchange BBO for market information.	178
4.17	Kurtosis as a function of timescale. Results are shown for the <i>base2x</i> and <i>diverse2x</i> agent configurations under the NBBO and BBO information regimes.	180
4.18	Linear autocorrelation of the 1-minute returns for the <i>base2x</i> and <i>diverse2x</i> agent configurations with different market information settings.	182
4.19	Autocorrelation of the absolute 1-minute returns for the <i>base2x</i> and <i>diverse2x</i> agent configurations with different market information settings.	184

4.20	Volume/volatility correlation results for the <i>base2x</i> and <i>diverse2x</i> agent configurations with different market information settings.	186
A.1	Linear autocorrelation of the raw and normalized returns.	216
A.2	Skew of the return distributions.	216
A.3	Autocorrelation of absolute returns.	216
A.4	Leverage effect $L(\tau, \Delta t) = \text{corr}(r(t, \Delta t), r(t + \tau, \Delta t))$ for the raw and normalized returns.	217
A.5	Share volume versus lagged volatility.	217
A.6	Average 1-minute volatility versus lagged 30-minute volatility.	217
A.7	Asymmetry effect — Difference between positive and negative lags for the correlation between the average 1-minute volatility and lagged 30-minute volatility.	218
A.8	Linear autocorrelation of returns with theoretical versus bootstrap 95% confidence intervals.	219
A.9	Side-by-side comparison of the autocorrelation of trade volume and absolute returns at 1-minute timescale.	220
A.10	Volume versus lagged volatility: $V(\tau, \Delta t) = \text{corr}(\text{volume}(t, \Delta t), r(t + \tau, \Delta t))$	221
A.11	Asymmetry in volume/volatility correlation: $D_V(\tau, \Delta t) = V(\tau, \Delta t) - V(-\tau, \Delta t)$, with $\Delta t = 1$ min.	221
A.12	Percentage of normalized returns which were losses above a quantile-cutoff.	222
A.13	Fano factor of the number of extreme returns that arrive within a coarse period of time.	224
A.14	Kurtosis of the distribution of interarrival times of extreme returns.	224
A.15	Leverage effect with squared normalized returns as the volatility measure: $L(\tau, \Delta t) = \text{corr}(r(t, \Delta t), (r(t + \tau, \Delta t))^2)$	225
A.16	Raw kurtosis as a function of timescale for the raw clock-time returns. Solid lines show the kurtosis for a symbol's full sample of intraday returns. Shaded regions show the 95% confidence interval from bootstrapping.	227
A.17	Raw kurtosis as a function of timescale for the normalized clock-time returns. Solid lines show the kurtosis for a symbol's full sample of intraday returns. Shaded regions show the 95% confidence interval from bootstrapping	227
A.18	Raw kurtosis as a function of timescale for the normalized event-time returns. Solid lines show the kurtosis for a symbol's full sample of intraday returns. Shaded regions show the 95% confidence interval from bootstrapping.	228

A.19	Alternative visualizations of the autocorrelation of absolute normalized returns.	229
A.20	Asymmetry in timescales.	230
A.21	Asymmetry effect — Difference between positive and negative lags for the correlation between the average fine volatility and lagged coarse volatility.	230
A.22	Correlation between Rogers-Satchell volatility and lagged coarse volatility — normalized clock-time returns.	231
A.23	Asymmetry in timescales using RS-volatility — normalized returns.	232
A.24	Correlation between Rogers-Satchell volatility and coarse volatility — normalized trade-time.	232
A.25	Asymmetry in timescales using RS-volatility — normalized trade-time returns.	232
B.1	Execution time and spread results from our <i>MarketSim</i> runs of WWW experiments for Environments A and B , $N = 66$, and no market maker.	245
B.2	ZI order routing logic from <i>MarketSim</i>	246
B.3	Mean surplus results from our <i>BestGuess+MS</i> and <i>BestGuess+MS+bug</i> implementations of the model versus figures reproduced from WW.	250
B.4	Mean execution time results from our <i>BestGuess+MS</i> and <i>BestGuess+MS+bug</i> implementations of the model versus figures reproduced from WW.	251
B.5	Mean median spreads from the exchange BBOs for our <i>BestGuess+MS</i> and <i>BestGuess+MS+bug</i> implementations of the model versus the figures reproduced from WW.	252
B.6	Mean median SIP NBBO spread for our <i>BestGuess+MS</i> and <i>BestGuess+MS+bug</i> implementations of the model versus the figures reproduced from WW.	253
B.7	Mean number of transactions from our <i>BestGuess+MS</i> and <i>BestGuess+MS+bug</i> implementations of the model versus the figures reproduced from WW.	254
C.1	Surplus and execution time results for our replication of the 2016 Wah and Wellman model.	328
C.2	Spread and transaction results for our replication of the 2016 Wah and Wellman model.	329
C.3	Example DBBO mid-price and return series from single simulations of each WW environment.	331

C.4	Linear autocorrelation of returns from the WW model.	333
C.5	Autocorrelation of absolute returns for the WW model.	334
C.6	Kurtosis as a function of timescale for return distributions from the WW model.	336
C.7	Volume/volatility correlation for the WW model.	337
C.8	Relative absolute deviation (RAD) versus consolidated-market baseline without latency.	339
C.9	CCDFs of absolute return distributions for <i>base</i> and <i>diverse</i> agent configurations under the NBBO information regime.	340
C.10	CCDFs of absolute return distributions for <i>base</i> and <i>diverse</i> agent configurations under the BBO information regime.	341
C.11	Kurtosis as a function of timescale for the <i>base</i> and <i>diverse</i> agent configurations.	342
C.12	Linear autocorrelation results for the <i>base</i> and <i>diverse</i> agent configurations with different market information settings.	344
C.13	Absolute autocorrelation results for the <i>base</i> and <i>diverse</i> agent configurations with different market information settings.	346
C.14	Volume-volatility correlation results for the <i>base</i> and <i>diverse</i> agent configurations with different market information settings.	348

LIST OF TABLES

2.1	Breakdown of which stylized facts we found evidence for in clock-time and event-time.	31
2.2	Stats on the number of trades per day for Dow 30 stocks in our sample.	33
2.3	Fano factor of the number of extreme (99 th -quantile) returns that arrive within a coarse period of time.	46
3.1	ODD protocol entities for WW.	65
3.2	ODD protocol state variables for WW.	67
3.3	ODD protocol scales for WW.	68
3.4	Parameter settings for the three market environments from WW.	77
3.5	Default parameters used across model configurations.	78
3.6	ZI strategy combinations, taken from WW [189, p. 80].	79
3.7	False rejection rates for quantitative self-alignment of <i>BestGuess</i> ZI surplus values under two testing methodologies.	86
3.8	Test of quantitative alignment of ZI surplus results from our <i>BestGuess</i> replication versus WW.	91
3.9	Test of quantitative alignment of LA surplus results from our <i>BestGuess</i> replication versus WW.	92
3.10	Test of quantitative alignment of ZI surplus from <i>MarketSim</i> 's results versus WW.	95
3.11	Test of quantitative alignment of LA surplus from <i>MarketSim</i> 's results versus WW.	96
3.12	Test of quantitative alignment of ZI surplus from our <i>BestGuess+MS</i> implementation results versus WW.	103
3.13	Test of quantitative alignment of LA surplus results from our <i>BestGuess+MS</i> replication versus WW.	104
3.14	Qualitative summary of model behavior from WW versus our <i>BestGuess</i> results.	115
4.1	Parameters and defaults for the WW+GetReal experiments.	150
A.1	Kurtosis of the (intraday) interarrival times for 99 th -percentile extreme normalized returns.	223
A.2	Intraday interarrival times (in μ s) of trades for Dow 30 stocks in our sample.	233
A.1	Power-law fit and comparison versus alternative distributions for 1-minute normalized returns.	235

A.1	Power-law fit and comparison versus alternative distributions for 30-minute normalized returns.	236
A.1	Power-law fit and comparison versus alternative distributions for 60-minute normalized returns.	237
A.1	Power-law fit and comparison versus alternative distributions for trade-level normalized returns.	238
A.1	Power-law fit and comparison versus alternative distributions for 100-trade normalized returns.	239
A.1	Power-law fit and comparison versus alternative distributions for 1000-trade normalized returns.	240
B.1	Test of quantitative alignment of <i>MarketSim</i> against the results reported by WW.	244
B.2	Test of quantitative alignment for our <i>BestGuess+MS+bug</i> implementation results versus WW.	248
B.3	Test of quantitative alignment of LA surplus results from our <i>BestGuess+MS+bug</i> replication versus WW.	249
B.4	Entities in the model ODD protocol.	255
B.5	State variables by entity in the model ODD protocol.	256
B.6	Scales in the model ODD protocol.	257
B.7	Parameter settings for the three market environments from WW. . .	263
B.8	Default parameters used across model configurations. These were specified by WW.	263
B.9	ZI strategy combinations, taken from WW.	264
B.10	ZI surplus-maximizing equilibrium mixtures for each experiment (environment+configuration+latency).	265
C.1	Entities in the model ODD protocol.	281
C.2	State variables for the non-trader entities in the model ODD protocol.	282
C.3	State variables for the WW trader classes in the model ODD protocol.	283
C.4	State variables for the Get Real [187] trader classes in the model ODD protocol.	284
C.5	Scales in the model ODD protocol.	285
C.6	Parameter settings for the three market environments from WW. . .	290
C.7	Default parameters used across WW model configurations.	290
C.8	ZI strategy combinations.	291
C.9	ZIRP surplus-maximizing equilibrium mixtures for each experiment. . .	292
C.10	Parameters and defaults for the Value agents.	310
C.11	Parameters and defaults for the Market Maker agent.	313
C.12	Parameters and defaults for the Momentum agent.	315

C.13 Default parameters used across model configurations.	326
C.14 Parameter settings for the three market environments from WW. . .	326
C.15 ZIRP strategy parameter combinations.	327

CHAPTER 1

INTRODUCTION

If we were to make a list of the most important, the most beautiful laws of natural science that have been discovered in the past three centuries, we would see that the vast majority of them hold only approximately, and only if we are willing to ignore details of microstructure.

—Simon [163, p. 25]

There was a time when wise people thought planets should revolve in circles. When observation showed incontrovertibly that they did not, the question was asked, ‘Why not?’ People tried to figure what kept the planets from displaying perfect circles. In the end it was realized that, in accordance with the laws of motion and gravitation, there never had been any reasons to expect circles. Circles were not the norm; ellipses were.

—Schelling [155, p. 33]

Simon [165] provides a rough definition of a complex system as one in which the properties of the system cannot be trivially inferred from the properties of its parts

and the rules governing their interaction. Such systems are often characterized by a hierarchical structure with partial independence between levels [163, 165]. This allows for behavior at one level of the system to be at least partially understood while abstracting away the detail of lower levels [163]. Simon [163] warns, however, that such description of behavior will fall short of exactness because the influence of the underlying subsystems will ‘show through’.

In this dissertation we investigate high-level descriptions of behavior and their relationship to underlying structure in the context of the modern U.S. equity market, also commonly known as the National Market System (NMS). The NMS, and financial markets more broadly, provide a rich opportunity for studying the behavior of complex systems [111]. The system produces a wealth of high-quality, high-frequency quantitative data observed across multiple levels of its organization. Over time, these data have provided an extensive set of empirical observations and regularities of the NMS and markets more broadly [41, 54, 111]. The system is also well-organized, comprised broadly of traders, trading venues, and the regulators which establish and maintain market rules [173]. The interaction of these agents, each with their own incentives and constraints [162], may collectively act as an efficient social-coordinating mechanism [96] or yield undesirable aggregate outcomes [111, 155]. In either direction, the resulting characteristics of the aggregate need not resemble the intentions of individual participants but instead depend more holistically on the system of interaction between individuals and their environment [155].

In the following chapters, we detail three works that collectively improve our understanding of how aggregate behavior of the NMS relates to its underlying organization. In Chapter 2, we investigate whether previous descriptions of aggregate

behavior—specifically, price changes—are still supported in empirical data following regulatory, technological, and structural changes to the system. We then focus on an explicit feature of the NMS whose effects on aggregate dynamics have not been well-established—the fragmentation of trading across multiple venues. We begin in Chapter 3 by independently replicating a leading agent-based model (ABM) on the effects of latency arbitrage in a fragmented market. We then extend this model to examine whether fragmented versus consolidated structure in a simulated market changes the aggregate price dynamics.

1.1 BACKGROUND

The NMS was established in 1975 to link together the individual markets at which securities were traded in the U.S. [53]. We focus here on NMS stocks, which are a type of security representing an ownership interest—or equity—in a company [177]. Stocks are fungible in the NMS, meaning that two shares of the same stock are interchangeable [173]. Transactions in the market occur by matching together traders looking to buy and sell some number of shares of a stock.

The price of an asset, perhaps the most fundamental quantity produced by a financial market, is established through the collective actions of individual traders. It is typically measured from transaction prices or the prices at which traders express interest in buying or selling the asset. Relative changes in price are called returns, and most studies focus on log-returns [55], measured as the logarithmic difference between two prices:

$$r(t) = \log p(t) - \log p(t-1).$$

An example intraday price series and the corresponding log-returns are shown in Fig. 1.1 for a stock from the data sample used in our analysis in Chapter 2. See Section 2.2 for more detail on the data and our specific methodology for constructing these series.

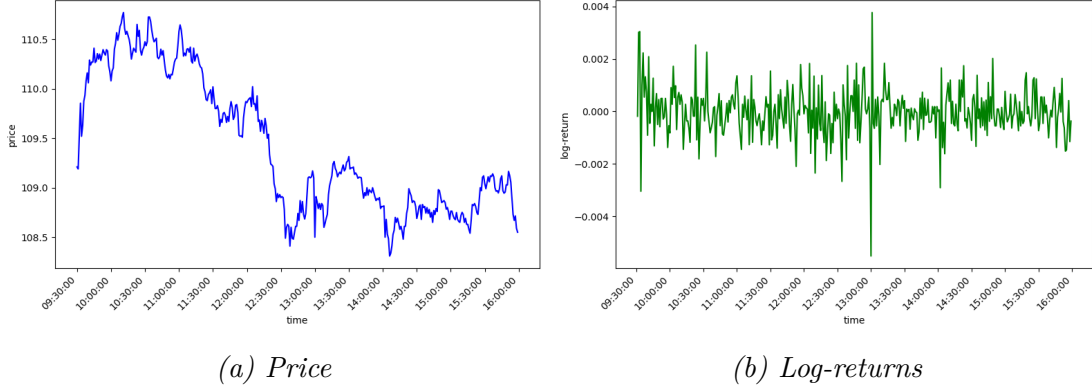


Figure 1.1: Example intraday price series and the corresponding log-returns for a Dow 30 stock from our data sample, detailed in Section 2.2. Price is sampled at one-minute intervals, using the last trade price from each minute as reported by the Securities Information Processor (SIP) to Thesys Group Inc., our data provider.

Traders placing orders, the resulting transactions, and the processes that record these actions as data must necessarily occur in specific locations and according to the rules implemented by the particular institution performing the matching process. The locations, rules, and mechanisms through which transactions occur in the NMS have changed over time, alongside corresponding social, economic, and technological changes [9, 184]. The market has become increasingly automated, with corresponding shifts in trading volumes and speeds, regulatory rules, and market structure [9, 106, 114, 184]. Thus, the means by which price is established for a stock in the NMS has also changed over time, raising questions about how this affects our understanding of price dynamics.

1.2 AGGREGATE PRICE DYNAMICS

A particularly influential strain of market modeling has attempted to describe the aggregate dynamics of price without reference to the market structure below. This class of models largely utilizes the high-level assumptions of a random walk, with a focus on the current price and the variability of its path over a future period [16, 142]. An economic rationale for such behavior is provided by the efficient market hypothesis (EMH), under which prices reflect available information so that returns cannot be systematically predicted [75, 154]. A random walk adds the stronger assumption that successive returns are independent and identically distributed. The random walk model and its related assumptions underlie some of the most prominent models in finance, such as the Black-Scholes-Merton option pricing formula [21, 130]. These developments helped establish the default expectations for the behavior of price in a market.

As data on prices observed in real-world markets have become increasingly available, departures from these expectations have been repeatedly documented [41, 54, 124]. Under a multiplicative random walk, the returns at one moment are independent of returns at the next; in its standard Gaussian form, it is further assumed that the log-returns are normally distributed. Mandelbrot et al. [124] in 1963 reported that large daily log-returns of cotton prices occur much more frequently than predicted under a normal distribution. This property is referred to as the return distribution exhibiting ‘heavy tails’.

Mandelbrot et al. [124] also noted that large changes in price tend to be followed by large changes, while small changes tend to be followed by small changes. This

property has come to be referred to as ‘volatility clustering’, with volatility broadly defined as the variability of the price. Visual evidence of this phenomenon is shown in the example log-returns in Figure 1.1, where periods of larger changes are separated by periods of relative calm. The study and modeling of volatility has become one of the most active branches of modern financial research [6, 23, 71, 153]. Much of the resulting work acknowledges and seeks to correct for the shortcomings of the aggregate, random-walk-based approach without fully abandoning it [77, 102].

In 2001, Cont [54] synthesized the empirical literature from the second half of the 1900s, producing a list of 11 ‘stylized facts’ of returns. These statistical properties—observed across many assets, markets, and time periods—provide a description of price behavior in markets as observed by numerous independent studies. We introduce Cont’s full list of stylized facts and briefly summarize the background literature for each property in Section 2.1.2. The emergence of common properties across many different market contexts is in keeping with Simon’s hypothesis that common aggregate behavior of a system can arise primarily from the organization of its parts rather than its specific contents [165]. Cont’s review is now widely cited, with numerous modelers using these statistical properties as empirical benchmarks¹.

These empirical findings were documented under particular market conditions—markets as they existed in the latter half of the twentieth century. As noted above, markets—and the NMS in particular—have undergone marked evolution in organization in the decades since Cont’s review [8, 9, 54, 106, 173]. We provide more detail on the recent evolution of the NMS in Section 2.1.3. This evolution raises the question of whether the properties described by Cont are robust to modern markets. Also, no

¹See Chen, Chang, and Du [44] and Chakraborti et al. [41] for reviews of agent-based models targeting these and other empirical regularities.

study surveyed by Cont examined the full set of 11 properties for a consistent set of assets, leaving it unclear whether the returns for a single asset or set of assets will express all of these properties at once. Answering these open questions is important in order to develop an understanding of empirical behavior in the system as it exists today and helps establish the empirical benchmarks that should inform the next generation of market models. Chapter 2 examines these questions in the context of the modern NMS, testing whether we can find evidence of Cont’s 11 stylized facts for a consistent set of stocks circa 2018–2019.

1.2.1 GENERATIVE MODELS OF AGGREGATE BEHAVIOR

The previous section detailed systematic departures from a class of models built on assumptions of random determinants of price, abstracting away the microstructure and behavior of individual participants in the market. Another limitation of this approach is that, as it ignores the underlying organization of the market as a determinant of aggregate dynamics, it cannot tell us how changes to the market may alter those dynamics. An alternative is to take a ‘generative’ approach: grow the aggregate regularity from the ground up beginning with the local interactions of individual agents [74]. In his ‘motto of generative social science’, Epstein contends: “If you didn’t grow it, you didn’t explain its emergence” [74, p. 43]. A theoretical grounding for this approach comes from Simon [162], who proposed that agents in a market or economy are boundedly rational. Under this view, agents use heuristics, respond to local information, and operate under constraints imposed by other levels of the system (e.g., exchange rules, regulations) [165]. The particulars of the agents and the rules and market structure governing their interactions therefore become the

fundamental determinants of aggregate behavior.

A common approach to generative social science is to build agent-based models (ABMs). These are computational models in which individual agents are instantiated in software with specified states and rules of behavior [15]. Their interactions can then be simulated and observed, providing a computational laboratory in which controlled experiments can be run under the modeled scenarios [15, 74]. An ABM provides a ‘sufficiency theorem’ by demonstrating that its assumptions are sufficient to produce a given result [74]. Generating an empirical regularity is a necessary condition for the model to be a candidate explanation of that regularity. If there are multiple candidate explanations, their agreement with empirical observation at other levels of the system’s structure can help determine which is more tenable empirically [74].

ABMs of financial markets have been around for decades, with a primary emphasis on generating the empirical regularities of markets from the interaction of heterogeneous, boundedly-rational agents [57, 99, 120, 144, 150]. Despite the increased flexibility and granularity available to this modeling approach, the models necessarily still must simplify and aggregate components of the referent system. Many of the early financial ABM efforts emphasized the ecology of trading agents and how specific strategies, or mixtures of strategies, can give rise to the empirical regularities of price dynamics in a stylized trading environment [57, 101, 119]. LeBaron’s 2001 guide to building agent-based financial markets [111] argues that the representation and structure of trading agents is the most important design decision, ahead of the specific trading mechanism.

Most early agent-based market models addressed price formation via aggregate excess demand or through an entity that computes a market-clearing price [57, 111,

118, 150]. An alternative route is to explicitly model a trading mechanism that mirrors real markets [45, 76, 149]. However, even the models that represent an explicit trading mechanism generally assume a single, consolidated exchange. Trading in the NMS is meanwhile highly fragmented, split across 18 stock exchanges and hundreds of off-exchange trading venues at the start of 2026 [40]. A model that explicitly represents a market as a consolidated exchange therefore increases the fidelity of its microstructure relative to stochastic process models or models calculating price from excess demand, but it still significantly simplifies market structure. The effects of this simplification on aggregate dynamics are largely unexplored in the literature.

A handful of models have attempted to fill this gap by introducing multiple exchanges and other structural features related to market fragmentation from the referent system. To our knowledge, Wah and Wellman [189, 190] provided the first ABM to examine the effects of latency arbitrage in a fragmented market. They began with a consolidated exchange before adding an additional exchange, communication latency, and a latency arbitrage agent. A major contribution of Wah and Wellman’s work was therefore the introduction of these specific, lower-level structural realities from the referent system to the field of financial ABMs. As a result, their studies have served as a cornerstone of subsequent models that have sought to incorporate similar features of the fragmented, latency-sensitive modern market. Subsequent models have introduced additional exchanges [66, 183], isolated the effects of latency [38], or incorporated other features of market structure from the real-world market [174, 175, 183].

These are important steps for developing an understanding of market dynamics generated from microspecifications, but there are still large gaps between market

models and markets in the wild. The primary regulator of the NMS, the Securities and Exchange Commission (SEC), has noted a current lack of market simulation frameworks representing the structural features they are focused on, while also expressing concerns about how trustworthy these tools are at producing robust insights for real-world markets [159]. We can address these gaps by rigorously verifying and empirically benchmarking the existing forefront of financial ABMs before then extending these models to incorporate greater complexity and fidelity to the referent system. Towards these ends, we undertook a multi-stage effort to replicate, empirically benchmark, and extend Wah and Wellman’s 2016 model [189]. Chapter 3 details our effort to independently replicate the model, revealing previously undisclosed nuances of the model and takeaways for the general process of replicating ABMs. In Chapter 4, we then extend the replicated model to test whether and how the represented features of fragmentation affect aggregate price dynamics. In both of these computational investigations, we document how changes to the information perceived and acted on by traders can materially affect aggregate outcomes in a market.

1.2.2 SYNTHESIS

Cont argues that the return properties described by his stylized facts are “model free in the sense that they do not result from a parametric hypothesis on the return process but rather from general hypotheses of qualitative nature” [54, p. 233]. However, the history above shows that these properties were identified in the context of the dominant models that have been created to describe price behavior. The assumptions of those models produced predictions that could then be compared to behavior observed in the wild. Most of Cont’s stylized facts are departures from the dominant models,

and their prominence may owe in part to their anomalous nature relative to widely accepted theory. As in Schelling’s account of planetary orbits [155], quoted earlier in this introduction, what is labeled as anomalous may instead reflect an inappropriate default that was never implied by underlying structure.

These developments, along with advances in computational power and data availability, have motivated new approaches to modeling [112]. ABMs provide a computational laboratory from which we can incrementally discover candidate mechanisms for the regularities exhibited by complex systems [74]. These computational laboratories do not replace observation in the wild or other forms of experiments, but they provide a complementary tool that allows us to grow macroscopic behavior from microspecifications [74]. A robust and rigorous understanding of the regularities exhibited by a system are fundamental to this process. Microspecifications that represent realistic structure from the system can then push us further towards understanding the mechanisms underlying these regularities.

Models, by necessity, abstract and simplify the referent system and will therefore at best explain the empirical phenomena only approximately [165]. The robustness of our understanding of empirical regularities and their sensitivity to the specifics of the market’s fragmented structure are the focus of the chapters that follow: we first test which of the canonical regularities of price returns are supported in data from the modern NMS; then we replicate, extend, and analyze a model capable of generating aggregate behavior from fragmented structure.

CHAPTER 2

REVISITING CONT’S STYLIZED FACTS FOR MODERN STOCK MARKETS

ABSTRACT

In 2001, Rama Cont introduced a now-widely used set of ‘stylized facts’ to synthesize empirical studies of financial price changes (returns), resulting in 11 statistical properties common to a large set of assets and markets. These properties are viewed as constraints a model should be able to reproduce in order to accurately represent returns in a market. It has not been established whether the characteristics Cont noted in 2001 still hold for modern markets following significant regulatory shifts and technological advances. It is also not clear whether a given time series of financial returns for an asset will express all 11 stylized facts. We test both of these propositions by attempting to replicate each of Cont’s 11 stylized facts for intraday returns of the individual stocks in the Dow 30, using the same authoritative data as that used by the U.S. regulator from October 2018–March 2019. We find conclusive evidence

for eight of Cont’s original facts and no support for the remaining three. Fact #10 (volume/volatility correlation) is only supported when returns are measured in clock-time (based on timestamps) and not when returns are measured in event-time (with trades as the events). Our study represents the first test of Cont’s 11 stylized facts against a consistent set of stocks, therefore providing insight into how these stylized facts should be viewed in the context of modern stock markets.

2.1 INTRODUCTION

Financial markets feature numerous agents collectively establishing an evolving price for an asset. Despite the complex nature of this process over time and the specific details of the asset in question, scientists for decades have noted seeming commonalities that emerge across markets. These high-level characterizations are referred to as stylized facts, which are then used to inform and evaluate models of markets. One of the most, if not the most, widely attributed set of stylized facts of financial markets was written by Rama Cont and published in 2001 [54]. Cont’s review summarized research ranging over the decades prior, noting 11 seemingly common statistical properties of price changes (returns) across different markets and time frames. Cont argued these properties ‘should be viewed as constraints that a stochastic process has to verify in order to reproduce the statistical properties of returns accurately’ [54, p. 233].

Markets in the 21st century have experienced a marked shift through increasing automation and responding regulatory changes [8, 33, 106, 109, 173, 184]. It has not been established whether the characteristics Cont [54] noted still describe modern markets, nor whether the full set of stylized facts should be expected to hold for

individual securities for a given time period. In this study, we examine the most granular stock trading data publicly available over the time period of 18 Oct. 2018–19 Mar. 2019, testing whether intraday stock returns in the modern U.S. stock market express each of Cont’s 11 stylized facts. Our earlier conference paper [152] provided preliminary results for a convenience sample of 10 stocks. This study extends and supersedes that work, testing an expanded set of analyses for the full set of stocks in the Dow 30. As summarized in Table 2.1, we find clear support for eight of the stylized facts and no support for the other three from the measures tested. Fact #10 (volume/volatility correlation) is only supported when returns are measured in clock-time (based on timestamps) and not when returns are measured in event-time (with trades as the events). Section 2.1.1 provides further motivation and background for the study. Section 2.2 details our data and general methodology, and Section 2.3 gives our specific analyses and results for each of the stylized facts. Section 2.4 discusses our findings, possible future work, and takeaways from our results.

2.1.1 BACKGROUND

Cont’s stylized facts [54] drew from numerous past results of Cont and others over the second-half of the 1900s into the early 2000s. Some of the facts were demonstrated through results in the paper, others cited past results, and a couple (detailed in Section 2.1.2) do not appear to have been directly cited or reproduced in the paper. Cont [54] argued for these properties to be used as constraints a model should be able to reproduce, and in the years since they have been frequently used to benchmark the empirical relevance of agent-based models (ABMs) to real-world financial markets [12]. Cont himself was involved in these efforts [85], and numerous ABMs have

similarly replicated multiple stylized facts [25, 106, 127, 149, 175]. More recently, Katahira et al. [108] gave a ‘speculation game’ model with results reproducing 10 of the 11 facts (all except Fact #3: Gain/Loss Asymmetry). In Bookstaber, Foley, and Tivnan [24] and Tivnan et al. [175], the stylized facts were used to determine which parameters produce realistic return series. As Cont noted in 2001, ‘most currently existing models fail to reproduce all these statistical features at once’ [54, p. 233]. An implicit assumption is therefore that the stylized facts should each hold for a given return series.

Since Cont’s set of stylized facts drew from a variety of results and research groups, no single asset (exchange rate, stock, index, etc.) was used for all 11 facts [54]. The most facts tested by a single study that we have found is eight, done by Chakraborti et al. [41] in their review on econophysics. In their summary of stylized facts, they provided example results using the French stock BNPP.PA from 2007–2008, with evidence supporting Facts #1, 2, 4, 5, 6, 7, 8, and 10. Determining the extent each of the stylized facts should be expected to hold for a given asset is important to understand how the facts should be used in practice. If a given asset’s returns over some time period cannot reliably be expected to exhibit all 11 facts, alternative interpretations should be considered. Perhaps stocks on average will exhibit all 11 properties, in which case comparing the expected value of a model’s results to the expected value from empirical data could be more appropriate¹.

How the stylized facts are exhibited under different constructions of time is also important to establish. Clock-time, meaning time as measured by a timestamp or date, is most widely represented in the stylized facts literature. As noted by Chakraborti

¹This is the approach taken by Farmer, Patelli, and Zovko [76] to validate their ‘Zero-Intelligence’ model.

et al. [41], this view inherently involves sampling, and the number of trades in a unit of time can vary widely period-to-period and stock-to-stock (Table 2.2). Event-time, using trades as the event in our case, smooths this variability out, with one trade occurring per unit of time. An event-based view of time is also natural in the context of market simulations, making any differentiation between the stylized facts in clock-time versus event-time crucial to understand when benchmarking these models.

2.1.2 CONT’S STYLIZED FACTS

Below are the stylized facts as given in Cont [54]:

- (1) **Absence of autocorrelations:** ‘(Linear) autocorrelations of asset returns are often insignificant, except for very small intraday timescales ($\simeq 20$ minutes) for which microstructure effects come into play.’
- (2) **Heavy tails:** ‘The (unconditional) distribution of returns seems to display a power-law or Pareto-like tail, with a tail index which is finite, higher than two and less than five for most data sets studied. In particular this excludes stable laws with infinite variance and the normal distribution. However the precise form of the tails is difficult to determine.’
- (3) **Gain/loss asymmetry:** ‘One observes large drawdowns in stock prices and stock index values but not equally large upward movements.’
- (4) **Aggregational Gaussianity:** ‘As one increases the timescale Δt over which returns are calculated, their distribution looks more and more like the normal distribution. In particular, the shape of the distribution is not the same at different timescales.’

- (5) **Intermittency:** ‘Returns display, at any time scale, a high degree of variability. This is quantified by the presence of irregular bursts in time series of a wide variety of volatility estimators.’
- (6) **Volatility clustering:** ‘Different measures of volatility display a positive autocorrelation over several days, which quantifies the fact that high-volatility events tend to cluster in time.’
- (7) **Conditional heavy tails:** ‘Even after correcting returns for volatility clustering (e.g., via GARCH-type models), the residual time series still exhibit heavy tails. However, the tails are less heavy than in the unconditional distribution of returns.’
- (8) **Slow decay of autocorrelation in absolute returns:** ‘The autocorrelation function of absolute returns decays slowly as a function of the time lag, roughly as a power law with an exponent $\beta \in [0.2, 0.4]$. This is sometimes interpreted as a sign of long-range dependence.’
- (9) **Leverage effect:** ‘Most measures of volatility of an asset are negatively correlated with the returns of that asset.’
- (10) **Volume/volatility correlation:** ‘Trading volume is correlated with all measures of volatility.’
- (11) **Asymmetry in timescales:** ‘Coarse-grained measures of volatility predict fine-scale volatility better than the other way around.’

As framed by Cont’s facts, financial returns overall are heavy-tailed, not independent and identically distributed (iid), and characterized by correlations and clustering

in behavior. Price changes themselves are not predicted by any of the stylized facts. Magnitudes of changes, seen as measures of volatility, are found to have nontrivial correlations and relationships with previous behavior. The lack of clear persisting signal on the raw returns is detailed in Fact #1, measured as a lack of linear autocorrelation in returns. This fact is reproduced in Cont [54] for event-time returns of the stock KLM and for the USD/Yen exchange rate. Nonzero autocorrelation function (ACF) values at the first lag are found in these and many other intraday results in general, with possible explanations proposed such as the ‘bid-ask bounce’, nonsynchronous trading effects, and partial price adjustment [7]. The effect has been found to decay to roughly zero within 15 minutes by Cont and others [10, 55, 176].

Some nonlinear transformations of returns, such as taking their absolute or squared values, provide measures of the magnitude of price changes. These volatility measures are found to exhibit persistent positive autocorrelation, in contrast to the linear ACF just discussed. Cont, Potters, and Bouchaud [58] and Cont [55] found the absolute 5-minute returns of S&P 500 futures had ACF values starting above 0.1 and not going below zero for at least 100 lags. Similar results were found in Chakraborti et al. [41], Cont [56], Liu et al. [116], Liu, Gopikrishnan, Stanley, et al. [115], Mikosch and Stărică [131], and Müller et al. [137]. Explicit power-law fits are given for the decay of autocorrelation in squared and absolute returns in Cont, Potters, and Bouchaud [58], Liu et al. [116], and Liu, Gopikrishnan, Stanley, et al. [115]. Power-law decay of absolute autocorrelation implies volatility exhibits long memory, or is ‘long-range correlated’, and we would in that case expect the correlation to asymptotically go to zero as the lags increase [125].

The variability of returns is well-documented and leads to the second stylized

fact: return distributions' heavy tails. The main measures of this feature discussed by Cont [54] are the kurtosis and tail index. Kurtosis is the fourth standardized moment of a distribution. Positive, or 'excess', kurtosis indicates a distribution has a sharper peak and heavier tails than the normal distribution. Empirical price returns have exhibited excess kurtosis for timescales up to multiple days [5, 23, 54, 145, 148], decreasing slowly as timescale increases [4, 28, 41, 55, 58, 123, 137, 148]. The ability to produce excess kurtosis has frequently been used in the financial ABM literature since 2001 as evidence of the empirical relevance of market models [24, 85, 108, 127, 174, 175].

While kurtosis gives a comparison of a distribution against the normal distribution, it does not give a pure measure of tailedness and does not necessarily give a useful comparison of non-normal distributions with respect to one another [63]. Furthermore, depending on the actual shape of the distribution's tails, it may not be well-defined. Cont [54] defines the tail index as the highest moment that is finite for a distribution. A number of studies have characterized return distribution tails as exhibiting asymptotic power-law behavior, estimating a tail index between two and five [28, 54, 78, 88, 95, 100]. If the tail index is less than or equal to four, the kurtosis will tend to infinity as sample size grows. However, the prevalence of scale-free behavior in empirical systems is controversial and can be difficult to definitively establish [30]. Alternative distributional descriptions of returns' tail behavior (e.g., lognormal, stretched-exponential) have been proposed where all moments of the distribution would be finite, and in general it can be difficult to distinguish across these classes of distributions [3, 123].

The fourth stylized fact (aggregational Gaussianity) describes the shape of the

return distribution changing and becoming more like the normal distribution as the timescale increases. With the caveats given above, decreasing kurtosis and increasing tail index as timescale increases provide indications of this fact [4, 28, 41, 58, 55, 148, 123, 137]. Chakraborti et al. [41] demonstrated aggregational Gaussianity visually and showed the distribution of returns converging towards the normal distribution more quickly in event-time than in clock-time. They explained this as event-time correcting for some amount of volatility versus clock-time returns. Tail-heaviness decreasing but not necessarily disappearing through methods of volatility correction is summarized in Fact #7, ‘conditional heavy tails’. Bollerslev, Chou, and Kroner [23] detail the ‘kurtosis problem’ of fat tails remaining in the residuals after applying autoregressive conditional heteroskedasticity (ARCH)-type models to stock returns. Andersen and Bollerslev [4] normalized 5-minute DEM/USD FX returns by an estimate of the average daily volatility pattern and reduced the kurtosis from 21.5 to 15.8. In Andersen et al. [5], the realized standard deviation of 5-minute returns was used to normalize daily returns of Dow 30 stocks, resulting in nearly normal kurtosis for the resulting distributions.

Taken together, returns’ heavy tails and volatility clustering lead to the characteristic that returns irregularly display periods of high volatility separated by long periods of relative calm. This ‘intermittency’ is Fact #5, and in much of the literature it is discussed as being visibly apparent in the returns series [54, 108] or following directly from these other facts. Cont [54] discusses the multifractal model as a possible explanation of intermittency, suggesting possible multiplicative processes operating across multiple timescales. Arneodo, Muzy, and Sornette [10] provide evidence of this, arguing for a multiplicative cascade of information from coarse timescales to

finer timescales. Müller et al. [138] and Zumbach and Lynch [198] presented evidence of this ‘asymmetry in timescales’ effect² (Fact #11) by measuring the correlation between fine volatility and lagged coarse volatility. These studies used different measures of volatility, but similarly found the coarse volatility predicted fine volatility better than the other way around. In similar analysis, Gencay, Selcuk, and Whitcher [84] found low volatility at a long timescale was likely followed by low volatility at a shorter timescale whereas high volatility did not necessarily show this same ‘vertical dependence’.

Fact #9, the ‘leverage effect’, expects a negative relationship between volatility and returns. Citing results from Bouchaud, Matacz, and Potters [27] and Pagan [143], Cont [54] specifically describes this effect as showing a negative correlation between returns and subsequent volatility, suggesting negative returns lead to increased volatility. Correlation of volatility with subsequent returns was negligible, meanwhile. Taken with Fact #11, the leverage effect and asymmetry in timescales have motivated the search for statistical models [22] that can capture ‘time-reversal asymmetry’ (TRA): financial time series do not behave the same forwards as they would if played in reverse [198]. Chicheportiche and Bouchaud [47] provided empirical evidence of asymmetry in timescales and the leverage effect using the daily prices of 280 stocks from 2000–2009. They showed general quadratic autoregressive (QARCH) models were able to capture TRA but overestimated the effect. Blanc, Donier, and Bouchaud [22] introduced Quadratic Hawkes models that displayed TRA, calibrating their model on 5-minute returns of 133 stocks from 2000–2009. They found evidence for the Zumbach effect in the empirical data but did not find a significant leverage

²Asymmetry in timescales (Fact #11) has also been referred to as the Zumbach effect [70] for the findings in Zumbach and Lynch [198].

effect at this timescale. Variations of the leverage effect have also been noted elsewhere in the literature, and negative (or even nonzero) correlation between returns and observed volatility is not always found [2].

According to Fact #10, volatility is positively correlated with trading volume. Clark [48] noted this relationship as far back as the 1970s when examining cotton prices. The relationship has been measured over the years through various means, including taking the correlation between the number of shares traded and absolute returns over a period of time [103] and measuring return variance as a function of the number of trades [41, 110, 145, 161]. If orders were arriving independently and at a constant rate in a market, the number of trades for a given time period should follow a Poisson process [41, 161]. Instead, trading volume overall has been shown to share a number of properties with the behavior of volatility. Financial markets notably have a daily pattern of trading and volatility characterized by high activity towards the ends of the day and relatively low activity in the middle of the day [1, 4, 60, 72, 123, 197]. Trading volume has been shown to exhibit heavy-tails and clustering similar to that found for volatility [103, 145]. Plerou et al. [145] proposed that the long-range autocorrelation of trading volume leads to volatility clustering (Fact #6). These and related results identifying strong autocorrelation in inter-trade durations have spurred the adoption of non-Poissonian point processes, notably Autoregressive Conditional Duration (ACD) models [72, 73] and self-exciting Hawkes processes [17, 29], to model the time-clustering nature of financial activity.

Finally, ‘gain/loss asymmetry’, the 3^{rd} stylized fact, is perhaps the least clear to interpret from the detail given by Cont [54]. This was summarized as larger draw-downs being seen than upward movements for stock prices and index values. It is

possible this is referring to results given in Cont [54] showing negative skewness in the return distributions for S&P 500 futures, Dollar/DM Futures, and Dollar/Swiss Franc futures, each at 5-minute timescales. The skew of a distribution implies something slightly different from the fact as summarized by Cont, however, as it does not necessarily tell you anything about which tail has larger values. Other studies have also found positive skew rather than negative [160]. Some of the literature since Cont [54] has examined ‘gain/loss asymmetry’ from another direction, looking at the amount of time it takes to see a gain versus a loss above a certain magnitude. This ‘inverse statistic’ has been used to show that a stock index will typically achieve a loss more quickly than a gain of the same magnitude [104], but the same property was not found for individual stocks [105], with correlated downward movements across stocks proposed as an explanation for why the phenomenon could arise in indices [166].

2.1.3 THE MODERN U.S. STOCK MARKET

It is possible that some of the stylized facts no longer hold much descriptive power due to structural changes to modern markets. The U.S. stock market, known as the National Market System (NMS), has had numerous regulatory and technological changes this century. Trades in the NMS occur through the matching of buyers and sellers of a stock at a given price point. This can occur on stock exchanges or off-exchange through brokers, peer-to-peer trading, or at Alternative Trading Systems (ATSs). Each step in this process is increasingly automated, from order submission by the trader to matching and execution on the trading venue. In response to this development, the Securities and Exchange Commission (SEC) in 2005 adopted Regulation NMS, a new set of rules aimed at modernizing the U.S. stock market and

providing increased protections for automated orders. As a result, the market has become more connected and more fragmented, with much higher trading volumes overall but a smaller portion of trades being executed by the stock exchanges than at the turn of the century.

As of 2018, there were 13 stock exchanges split across four geographic locations in northern New Jersey, and five more exchanges have been added by July 2026 [40]. If we include ATSS and over-the-counter (OTC) firms, the number of venues totals in the hundreds [80, 82]. Per data reporting requirements from Regulation NMS, messages and trades from this array of venues are consolidated by the three Security Information Processor (SIP) ‘tapes’. Exchanges and off-exchange venues must report trades to the SIP tape corresponding to the traded security based on its listing exchange. This is diagrammed at a high level in Figure 2.1. Roughly 30% of trades in our data occurred off-exchange, a portion that has increased to roughly 45% today [40]. Tivnan et al. [173] give a more detailed summary of the market’s infrastructure circa 2016, with evidence of impact from market fragmentation on the prices acted on by traders.

Each type of trading venue has its own specific rules and data-reporting requirements, and each venue reports trades with some (small but nonnegligible) allowed latency³. Systems executing automated orders are required to sync their clocks to within 50ms of the time maintained by the National Institute of Standards and Technology (NIST) in the US [117]. This may seem like a short amount of time, except the NMS is increasingly operating at speeds approaching the speed of light, with reaction times measured in microseconds or even nanoseconds [8]. Driven by algorithmic

³Bartlett III and McCrary [19] found trades of Dow 30 stocks were processed by the SIPs 24ms on average after they were recorded by exchange matching-engines over the period of 6 Aug. 2015–30 Jun. 2016. Off-exchange trades, meanwhile, are allowed up to 10 seconds to report to Trade Reporting Facilities (TRFs) which then report the trades to the SIPs [81].

trading, a stock price can have an extreme rise or drop and rebound back to nearly its original level in less than a blink of the eye [106]. Andersen et al. [5] noted the median time between trades was 23.1 seconds on average for Dow 30 stocks in 1993–1998. For contrast, in our sample, the median inter-trade duration for Dow 30 stocks was $694.03\mu\text{s}$ on average⁴, a reduction of more than four orders of magnitude.

Mechanisms have also been implemented in the decades since Cont’s study [54] to constrain the magnitude of price movements. The SEC introduced Limit Up-Limit Down (LULD) rules in 2013 to limit rapid price fluctuations for individual stocks [136]. At a high level, LULD bands prevent trades in NMS stocks outside a specified percentage level above or below a reference price calculated for each stock. For the most liquid (Tier 1) stocks priced above \$3.00, these bands are set at 5% above and below the previous 5-minute average price, except during the last 25 minutes of the trading day when the bands move to 10%⁵. In prior eras of the stock market, much larger swings were possible, as evidenced by the ‘Flash Crash’ of May 6, 2010, that prompted the LULD policy change. During that event, hundreds of securities over a 20-minute period executed trades at prices more than 60% below their respective prices at the start of the period [136]⁶. Thus, the tails of intraday return distributions are constrained now in ways they were not in the era studied in Cont [54]. Given these modern developments, it is vital to understand whether the high-level characteristics of the market observed by Cont and others circa 2001 still hold in the market we have today.

⁴Inter-trade durations are broken down in Table A.2 in our supplementary material.

⁵See www.luldplan.com for more detail.

⁶These trades were subsequently rolled back by the exchanges [136].

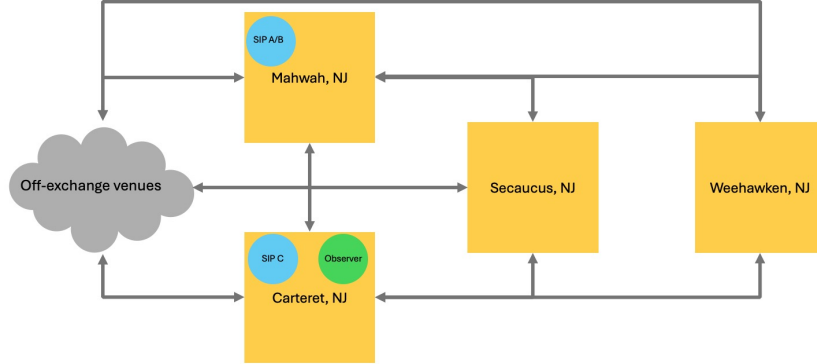


Figure 2.1: The U.S. National Market System (NMS) circa 2018. The data collected here was collected from the SIP feeds by Thesys, the ‘observer’ in Carteret. Image recreated from Van Oort et al. [184] and reproduced from Ratliff-Crain et al. [152] with permission from IEEE.

2.2 METHODOLOGY

We test each of Cont’s 11 stylized facts on U.S. stock data from 18 Oct. 2018–19 Mar. 2019. We removed two dates where the stock market closed early, Black Friday (23 Nov. 2018) and Christmas Eve (24 Dec. 2018), in order to have trading days of consistent length across our sample. This leaves us with 101 trading days for each stock. We specifically looked at the individual stocks included in the Dow Jones Industrial Average as of October 2018⁷ [65]. The Dow Jones Industrial Average, commonly referred to as the Dow 30, is a well-known index dating back more than 100 years. The index consists of an evolving set of 30 stocks claimed to represent all industries in the U.S. stock market except for transportation and utilities. The Dow

⁷The list of symbols is: AAPL, AXP, BA, CAT, CSCO, CVX, DIS, DWDP, GS, HD, IBM, INTC, JNJ, JPM, KO, MCD, MMM, MRK, MSFT, NKE, PFE, PG, TRV, UNH, UTX, V, VZ, WBA, WMT, XOM.

30 is widely used as an indicator of the economy’s performance more broadly, and the index and its underlying stocks are frequently used in studies of the stock market [5, 19, 104, 123, 173]. Thus, while this group of stocks is somewhat arbitrary, it gives an established cross-section of highly-traded stocks in the NMS.

Our data contain all trades reported for these symbols over the date range. The data were provided by Thesys Group Inc., which acted as a tape consolidator in the NMS and was the sole data provider for the SEC’s MIDAS in this time period [173]. The Thesys data were collected in the Nasdaq data center in Carteret, NJ (Figure 2.1). Due to the latency and limitations on clock synchronization mentioned in Section 2.1.3, the exact order of trades in the data is not definitive⁸, but our time series provide the perspective of an observer located in Carteret, NJ, viewing the events of the market as reported by the SIPs.

We consider two different views of time when constructing our return time series: clock-time based on timestamp and event-time, with trades as the event. In either view, in order to aggregate to any level that is less granular than a single trade per unit of time, we define $P(t, \Delta t)$ as the price of the last trade to occur in that time period. In clock-time, time points with no trades will be assumed to have the same price as the previous time point that had at least one trade, as no new price information has been received since then. Let $X(t, \Delta t) = \log P(t, \Delta t)$, the log-price. The log-return at time t and timescale Δt is defined as $r(t, \Delta t) = X(t, \Delta t) - X(t - 1, \Delta t)$. Any reference to returns going forward should be interpreted as meaning log-returns. Note that time points with no trades will have a return of 0. We measure volatility as the absolute returns $|r(t, \Delta t)|$.

⁸And, in fact, a definitive ordering is not possible due to special relativity [8, 69, 173].

We limit our time series to the trading day (9:30am–4:00pm ET), filtering out ‘after-hours’ trading activity. We also filter out the batch auctions that start and end each day. In our price time series then, the last price before the close from one trading day will be followed immediately by the first price after the open of the next trading day. Due to this construction, an overnight return between $t_1 = 16:00$ on a given day and $t_2 = 9:30$ the following day is characteristically different from a return between two sequential prices within the same trading day. Therefore, we only consider returns $r(t, \Delta t)$ such that t and $t - \Delta t$ are within the range 9:30–16:00 of the same trading day. If the last period of any day is incomplete (e.g., $\Delta t = 50$ min would result in a 40-minute period at the end of the day), we remove that return from our series. It is worth noting that the number of data points in a day will vary inversely with the timescale. Most stocks will have tens if not hundreds of thousands of trades per day, whereas there are 390 minutes in a 6.5-hour trading day.

As noted in Section 2.1.2, financial markets are commonly characterized by a daily ‘u-shaped’ pattern of relatively high activity at the beginning and end of the trading day [4, 60, 123, 197]. Consistent with many of the results cited here (e.g., Blanc, Donier, and Bouchaud [22], Malevergne, Pisarenko, and Sornette [123], Müller et al. [138], Muzy, Delour, and Bacry [139], and Zumbach and Lynch [198]), we correct our returns to account for this intraday seasonality before assessing the stylized facts. We focus on two methods of normalization. Event-time fixes the number of trades per unit of time, therefore correcting for varying activity levels [41]. In clock-time, we can normalize a return at time t by the average volatility at that time across the sample. More specifically, say that $r_T(t, \Delta t)$ indicates a return is from time-point t on day T . We first center and standardize $r_T(t, \Delta t)$ by the mean and standard deviation of

returns on that day at the given timescale:

$$r'_T(t, \Delta t) = \frac{r_T(t, \Delta t) - \langle r_T(\cdot, \Delta t) \rangle}{\sigma(T, \Delta t)}.$$

We do this standardization for both the event-time and clock-time returns. Next, for a clock-timescale Δt , let

$$v(t, \Delta t) = \langle |r'_T(t, \Delta t)| \rangle_{T=1,2,\dots,101},$$

the average absolute return at time t across the 101-day sample for the stock, and let

$$\hat{r}_T(t, \Delta t) = \frac{r'_T(t, \Delta t)}{v(t, \Delta t)}.$$

We do not do this latter normalization in event-time, since event-time is already correcting for different activity levels at different times of day by fixing the number of trades in a bucket. In other words, for an event-timescale N , $\hat{r}$ is simply:

$$\hat{r}_T(t, N) = r'_T(t, N).$$

We use these two views of normalized returns for most of the results in this study. The exception is our discussion of heavy tails (Section 2.3.2), which explicitly compares the raw (unnormalized) and normalized return distributions. The high-level takeaways for the other facts were largely consistent when looking at the raw returns, however, and we report those results in our supplementary materials (Appendix A.1).

Consistent with many of the results and references given by Cont [54], the main tools used in our analysis are correlation, the calculation of moments, and describ-

ing the distributions of events. We utilize Pearson sample correlation, denoted as $\text{corr}(x, y)$ going forward, to calculate the correlation values. Please note, we do not allow lagged correlations to cross day boundaries; for example, $\text{corr}(\hat{r}_T(t, \Delta t), \hat{r}_T(t + \tau, \Delta t))$ does not consider pairs where $t + \tau$ goes past the end of trading day T ⁹. In order to judge what is a clear, nontrivial feature of the returns in our sample, we look at the extent a result shows a consistent signal across symbols and the range of possibilities in 95% confidence intervals of the measure. We utilize moving block bootstrapping (MBB) methodology to construct the confidence intervals in order to account for the heavy-tailed, time-dependent structure of the data (as expected by many of the stylized facts). Our specific bootstrapping methodology is as follows: Let n_i denote the number of returns on date i for a given symbol and timescale. We calculate the block length as $L_i = \lfloor n_i^{1/3} \rfloor$ for this date. A block is then $B_j = [r_i(t_j, \Delta t), r_i(t_j + 1, \Delta t), \dots, r_i(t_j + L, \Delta t)]$ for $j \in [1, n_i - L_i]$. To construct a bootstrap sample, we randomly select, with replacement, $\lceil n_i/L_i \rceil$ blocks for each date i in our sample. The 95% confidence interval for a given measure is then the 2.5-percentile and 97.5-percentile values of the measure calculated on 1,000 of these samples.

2.3 RESULTS

Before getting into the specific results for each fact, Table 2.1 gives a high-level summary of our findings. We overall find strong evidence for 8 of the 11 stylized

⁹Going forward, we resume denoting returns generically as $r(t, \Delta t)$ when detailing various metrics. We specifically note which results are from the raw versus normalized returns.

Fact #	Fact Name	Clock-time	Event-time
1	Lack of linear ACF	X	X
2	Heavy tails	X	X
3	Gain/Loss asymmetry		
4	Aggregational Gaussianity	X	X
5	Intermittency	X	X
6	Volatility Clustering	X	X
7	Conditional heavy tails	X	X
8	Slow decay of abs. ACF	X	X
9	Leverage effect		
10	Volume/volatility corr.	X	
11	Asymmetry in timescales		

Table 2.1: Breakdown of which facts we found evidence for in clock-time and event-time. ‘X’ marks indicate strong evidence found for a fact in the respective timescale.

facts. However, when viewing the returns in event-time we only find evidence for 7 of the 11 facts, with no support found for Fact #10, volume/volatility correlation. Nuances to these results are unpacked in detail in the below subsections.

2.3.1 LINEAR AUTOCORRELATION OF RETURNS

Stylized Fact #1 expects that linear autocorrelations of returns are ‘often insignificant, except for very small intraday timescales ($\simeq 20$ minutes) for which microstructure effects come into play’ [54, p. 224]. The linear autocorrelation function (ACF) is

$$C(\tau, \Delta t) = \text{corr}(r(t, \Delta t), r(t + \tau, \Delta t)).$$

Figure 2.2 shows our linear autocorrelation results for the 1-minute and trade-level normalized returns. ACF is negative at the first lag in each set of results. The negative first-lag ACF for the normalized trade-level returns, ranging between -0.35

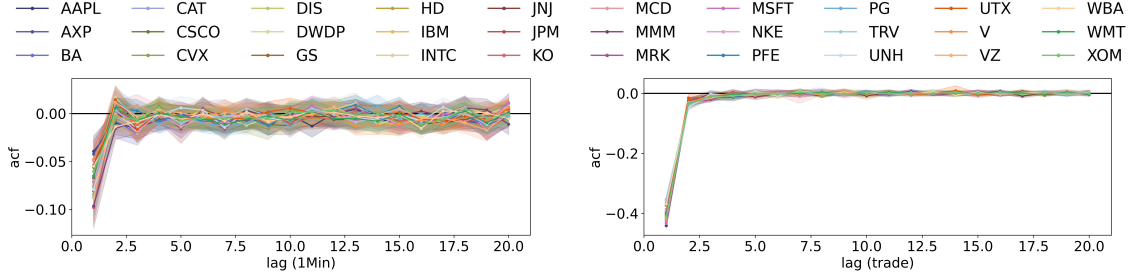


Figure 2.2: Linear autocorrelation of the normalized returns. Solid lines show the ACF for a symbol's full sample of intraday returns. Shaded regions show the 95% confidence interval from bootstrapping. Please note, the color legend shown here is used for all result plots.

and -0.5, is characteristic of the so-called ‘bid-ask bounce’ [7, 54]. Autocorrelation decreases towards zero in magnitude past the first lag. The sign of the ACF varies by symbol starting at lag $\tau = 2$ in clock-time and $\tau = 3$ in the trade-level results. The 95% bootstrap confidence intervals include zero for at least some symbols starting at these lags as well. We therefore argue these results show linear dependence is unpredictable and weak past the first couple lags.

2.3.2 HEAVY TAILS AND AGGREGATIONAL GAUSSIAN-ITY

Cont’s 2nd fact expects return distributions to exhibit heavy tails. Cont [54] specifically compares the heavy tails of the ‘unconditional’ distribution of returns (Fact #2) to returns corrected for volatility clustering (Fact #7). We therefore present results for both the raw clock-time returns along with our two sets of normalized returns here. As in Cont [54], we first look at the fourth central moment of returns, kurtosis, before discussing distributional fits and tail index.

symbol	mean	min	max	total
AAPL	253,830.06	117,999	721,853	25,636,836
AXP	33,788.66	19,789	73,697	3,412,655
BA	62,931.61	25,188	413,938	6,356,093
CAT	50,951.89	24,837	182,012	5,146,141
CSCO	121,011.00	68,021	263,445	12,222,111
CVX	52,289.07	32,564	94,161	5,281,196
DIS	63,146.17	35,996	193,100	6,377,763
DWDP	71,755.24	41,153	167,513	7,247,279
GS	40,314.43	16,930	118,901	4,071,757
HD	51,800.75	27,976	126,587	5,231,876
IBM	48,499.73	23,467	160,827	4,898,473
INTC	139,519.63	75,423	393,166	14,091,483
JNJ	66,037.43	36,204	412,483	6,669,780
JPM	111,506.73	66,703	253,560	11,262,180
KO	75,112.89	45,001	242,201	7,586,402
MCD	36,653.29	18,526	102,455	3,701,982
MMM	27,448.69	17,108	78,667	2,772,318
MRK	74,534.65	46,322	129,195	7,528,000
MSFT	235,902.65	117,100	539,691	23,826,168
NKE	48,406.32	26,397	154,043	4,889,038
PFE	105,805.89	62,119	200,288	10,686,395
PG	69,804.29	36,867	164,793	7,050,233
TRV	17,153.75	10,601	29,834	1,732,529
UNH	43,087.90	23,567	101,063	4,351,878
UTX	42,593.85	21,438	129,023	4,301,979
V	76,776.05	46,552	150,305	7,754,381
VZ	87,286.66	43,486	186,526	8,815,953
WBA	46,180.90	21,109	105,880	4,664,271
WMT	62,920.26	35,367	127,323	6,354,946
XOM	83,677.95	51,635	166,697	8,451,473
all	76,690.95	10,601	721,853	232,373,569

Table 2.2: Stats on the number of trades per day for Dow 30 stocks in our sample.

Kurtosis

The kurtosis of a distribution is defined as:

$$K(\Delta t) = \frac{\langle (r(t, \Delta t) - \langle r(t, \Delta t) \rangle)^4 \rangle}{\sigma(\Delta t)^4} - 3,$$

where $\sigma(\Delta t)^2$ is the variance of the returns. Note that this definition of kurtosis subtracts three in order for the normal distribution to have a kurtosis of zero. Positive, or ‘excess’, kurtosis means a distribution displays a sharper peak and heavier tails than a normal distribution.

Similar to Cont and collaborators in Cont, Potters, and Bouchaud [58] and Potters, Cont, and Bouchaud [148], we plot the kurtosis as a function of $j\Delta t$, expecting positive $K(j\Delta t)$ that decreases as j increases. Our results for the raw clock-time returns, normalized clock-time returns, and normalized event-time returns are shown in Figure 2.3. As expected, we find excess kurtosis that decreases as a function of timescale in each set of results. The raw return kurtosis values for $\Delta t = 1$ min vary by multiple orders of magnitude depending on the symbol. We see very wide confidence intervals for the raw returns relative to the normalized returns, likely due to heteroskedasticity and intraday seasonality effects that are removed by our normalization methods. All symbols have

$$K(\Delta t = 1 \text{ min}) > K(\Delta t = 60 \text{ min}) > 0$$

for the raw returns. This is as expected by Fact #4, ‘aggregational Gaussianity’, with past results finding return kurtosis decreased with timescale but stayed positive for

timescales up to multiple days [58, 137, 148].

Fact #7 expects that tail heaviness is reduced but not necessarily removed after correcting returns for volatility clustering. As expected, each symbol's returns still exhibit excess kurtosis after both our methods of normalization, at least for the smallest timescales. Correcting the 1-minute returns for the average intraday volatility pattern greatly reduces the kurtosis and narrows the confidence intervals. The normalized trade-time returns meanwhile have very large kurtosis, ranging from roughly 10^3 to 10^4 . Note, however, that there are many trades per minute on average, ranging from approximately 44 for TRV to 651 for AAPL (Table 2.2).

Normalized return kurtosis decreases nearly monotonically as a function of timescale, demonstrating aggregational Gaussianity more consistently and clearly than the raw clock-time returns. In event-time, $K(N = 2500) < 1$ for all symbols. Note that the raw clock-time returns have excess kurtosis at $\Delta t = 60$ min. There are more than 2,500 trades in an hour on average for our lowest-traded stock (TRV, Table 2.2), and most other symbols have many times that. The average kurtosis of the normalized clock-time returns is below zero for $\Delta t \geq 25$ min.

Kurtosis therefore gives us evidence in support of Facts #2, 4, and 7: heavy tails, aggregational Gaussianity, and conditional heavy tails. However, some caution is necessary when using kurtosis as a measure of heavy tails, as discussed in Section 2.1.2. In particular, it does not give a pure measure of tailedness, it might not be useful in comparing two non-normal distributions to each other, and it may tend towards infinity if the distribution has a tail index of four or smaller [63]. As possible evidence of this latter point, we found very large kurtosis with wide bootstrap error ranges in the raw returns at the smallest timescales. Sample size effects might also come into

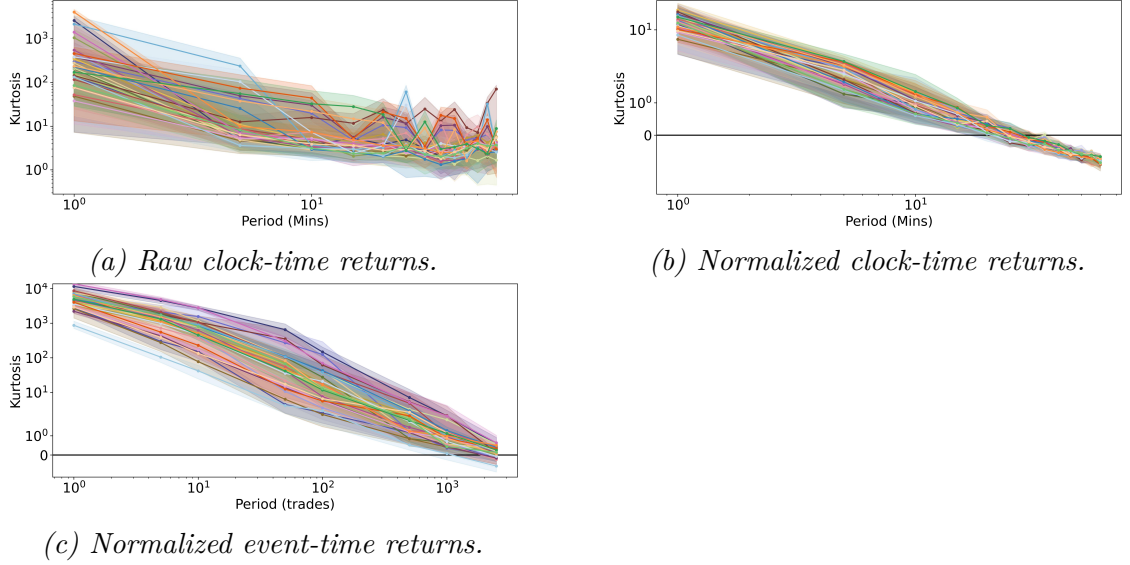


Figure 2.3: Kurtosis as a function of timescale. Solid lines show the kurtosis for a symbol's full sample of intraday returns. Shaded regions show the 95% confidence interval from bootstrapping.

play as the number of observations we have decreases as a function of timescale.

Tail index

We next try to estimate the tail index and test distributional fits to shed more light on how heavy the return distribution tails actually are. Cont proposes that ‘the (unconditional) distribution of returns seems to display a power-law or Pareto-like tail, with a tail index which is finite, higher than two and less than five for most data sets studied’ [54, p. 224]. To test this, we employ the well-established methodology introduced by Clauset, Shalizi, and Newman [49] to fit power laws for the tail distributions, estimate the tail exponent, and compare distributional fits. The probability density function for a continuous power law is:

$$p(x) = \frac{\alpha - 1}{x_{\min}^{-(\alpha-1)}} x^{-\alpha},$$

where $\alpha > 1$ and $x_{\min} > 0$ is a lower bound. The complementary cumulative distribution function (CCDF) is then:

$$P(x) = \Pr(X \geq x) = \int_x^\infty p(x') dx' = \left(\frac{x}{x_{\min}}\right)^{-(\alpha-1)}$$

Clauset et al. utilize the below maximum likelihood estimator (MLE), equivalent to the estimator introduced by Hill [98]:

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

for $x_{i \in [1, n]} \geq x_{\min}$. Iteratively calculating $\hat{\alpha}$ for different values of $x_{\min}$, the best fit is the $\{x_{\min}, \hat{\alpha}\}$ pair that minimizes the Kolmogorov-Smirnov (KS) statistic—the maximum distance between CCDFs of the empirical distribution and the fitted model. Assuming a good fit, the estimated tail index is then $k = \hat{\alpha} - 1$. Alternative distributional fits (e.g., exponential, lognormal) can also be tested against the power-law model in case they are more appropriate for the selected range $x \geq x_{\min}$, with implications for the tail index.

To determine the best power-law fit for each return distribution tail and compare alternative distributions, we utilize the **powerlaw** python library from Alstott, Bullmore, and Plenz [3], which implements the Clauset, Shalizi, and Newman [49] framework. We fit up to one million¹⁰ of the largest positive returns for a given timescale to evaluate the right tail. We employ the same procedure to look at the left tail, except we first multiply the returns by -1. Distributional comparisons are based on the relative likelihoods of the data under two candidate distributions, as

¹⁰This cutoff only affects the smallest event-timescales, as the clock-timescales do not have more than 1,000,000 data points per symbol.

detailed in Alstott, Bullmore, and Plenz [3] and Clauset, Shalizi, and Newman [49]. We choose one distribution over another if the log likelihood ratio favors it and the p-value is below 0.05; otherwise we say the test is inconclusive¹¹.

The estimated $\hat{\alpha}$ fits are plotted as a function of timescale in Figure 2.4. The raw 1-minute return tails each have $\hat{\alpha} \in [3, 5]$. This is consistent with prior estimates of a tail index $k = \alpha - 1$ in the 2–5 range [28, 54, 78, 88, 95, 100]. The range of $\hat{\alpha}$ values increases with timescale, as does the uncertainty of the estimate due to less data¹². Many of the tail-ranges selected for $\Delta t = 60$ min include fewer than 100 returns, making the fits rather uncertain (e.g., symbol UTX’s left tail has just 11 extreme returns included, with $\hat{\alpha} = 5.65 \pm 2.75$). Even so, most $\hat{\alpha}$ values fall in the 3–6 range across timescales for the raw clock-time returns.

Normalizing the clock-time returns by the daily volatility pattern increases $\hat{\alpha}$ on average, ranging from 4.58–6.03 at the 1-minute timescale. Aggregation greatly increases the range of $\hat{\alpha}$ values. Some larger-timescale tails have fits with $\hat{\alpha}$ lower than their respective exponent at $\Delta t = 1$ min, while others have $\hat{\alpha} > 10$, suggesting light tails and a poor fit for the power law. Increased timescale again comes with increased uncertainty, though, due to fewer returns in the tails. The normalized event-time tails have a tighter range of $\hat{\alpha}$ values for $N < 1000$. At trade level, the range of $\hat{\alpha}$ values is between 2 and 4, increasing to 4–6.5 for $N = 100$. At $N = 1000$, we again see some evidence of poor fits for the power-law model.

Based on the estimated $\hat{\alpha}$ values, moments greater than two, including skew and

¹¹Please note, the 0.05 p-value cutoff is somewhat arbitrary (Clauset, Shalizi, and Newman [49] use a cutoff of 0.1). The raw log likelihood ratio metrics r and p are reported in our appendix.

¹²The $\hat{\alpha}$, error margin, $x_{\min}$, and tail size values are reported at the timescale+symbol+tail-level in Appendix A.9. We do not use bootstrapping for the tail-fitting metrics. The error margin for $\hat{\alpha}$ is the 95th-percentile based on the theoretical standard error σ from Clauset, Shalizi, and Newman [49] and Alstott, Bullmore, and Plenz [3].

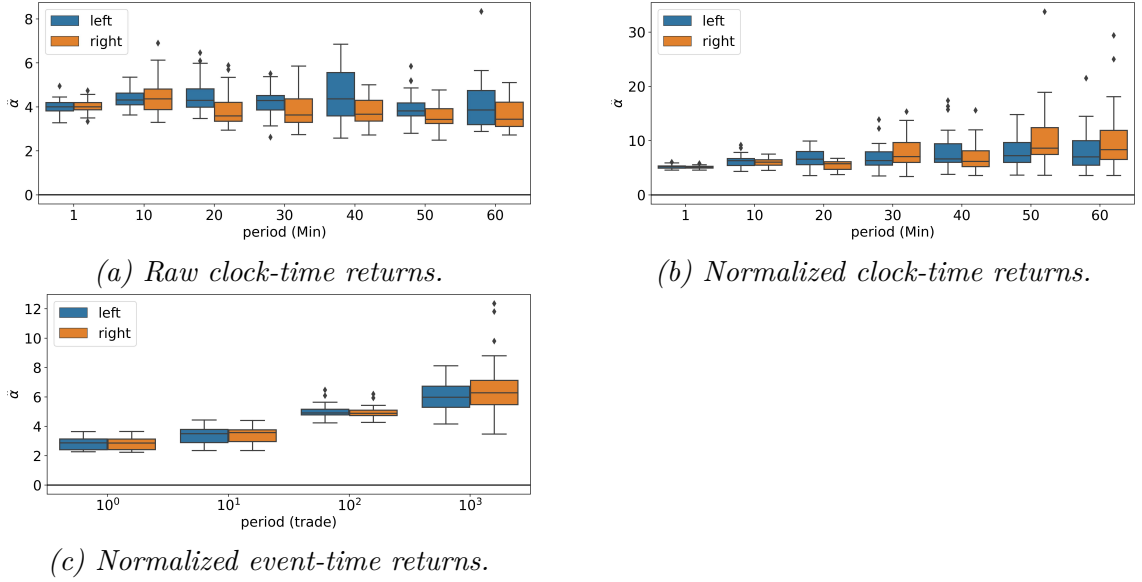


Figure 2.4: Box-and-whisker plot showing the range of power-law tail exponent $\hat{\alpha} = k + 1$ values estimated for the return distribution tails. Left versus right tails are plotted separately. Detailed metrics are provided for each power-law fit at the timescale+symbol+tail-level in Appendix A.9.

kurtosis, may need to be treated with caution for the raw clock-time returns. Even variance might not be finite for some of the normalized trade-level return distributions, assuming the power-law fit is appropriate. However, we must consider whether alternative distributions fit the data as well or better than the power law while having all their moments defined. The distributional comparisons are summarized in Figure 2.5. The log likelihood ratio test between the power-law and lognormal distributions was inconclusive or favored the lognormal in most cases for the raw 1-minute returns and for all raw clock-time returns when $\Delta t > 1$ min. Comparing between the power-law and exponential distributions favored the power law or was inconclusive in every case at $\Delta t = 1$ min. As timescale increases, we become more inconclusive when comparing against either distribution and even favor the exponential in some cases (albeit based on much fewer returns).

Normalizing the clock-time returns by the daily volatility pattern notably increases the portion of tails where we favor the exponential distribution when $\Delta t > 1$ min. The exponential distribution is preferred over the power law for almost half the tails at the 40-minute and 60-minute timescales. This makes sense, as excess kurtosis is largely removed for the normalized returns at these timescales (Figure 2.3). We find more evidence in support of the power law over the alternative distributions at the smaller event-time timescales. The power law is favored over the exponential distribution for every tail for $N \leq 10$. Most comparisons against the lognormal distribution are inconclusive at each event-time timescale. At $N = 1000$, some tails favor the exponential distribution over the power law. Excess kurtosis is greatly reduced by $N = 1000$ for the normalized event-time returns, although not as completely as the normalized clock-time returns.

To summarize, we have strong evidence of heavy tails at our base timescales. The raw and normalized 1-minute returns and the normalized trade-level returns each exhibit large positive kurtosis values, and we prefer the power law over the exponential distribution for almost every tail. The estimated tail exponents $\hat{\alpha}$ suggest a tail index between 2 and 5 for the 1-minute returns (raw and normalized) and between 1 and 3 for the normalized trade-level returns. However, the tail index interpretation depends on the tails behaving asymptotically like a power law. We do not rule out the lognormal distribution as a plausible fit compared to the power law for most tails in both timescales. Based on this and constraints on rapid price movements imposed in the empirical system (see Section 2.1.3), we cannot rule out all moments being finite. Aggregation and normalization both reduce kurtosis and increase $\hat{\alpha}$, consistent with the expectations of aggregational Gaussianity and conditional heavy

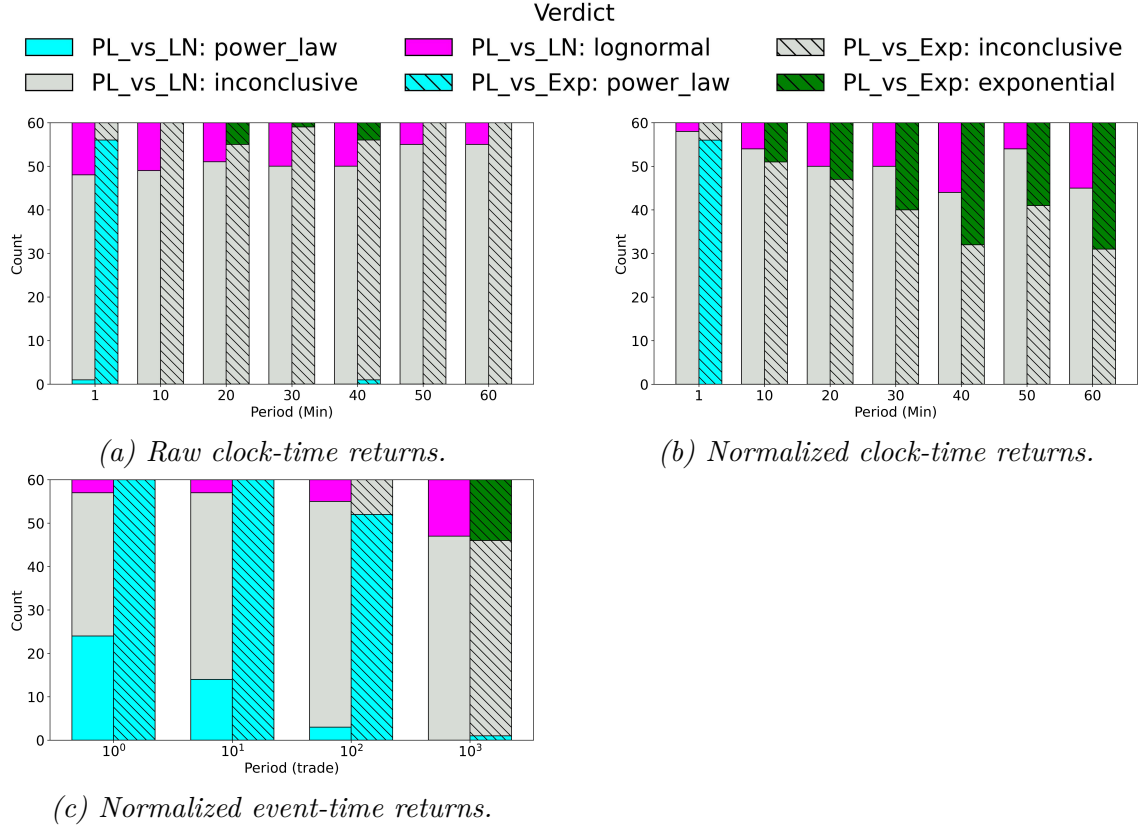


Figure 2.5: Stacked bar chart comparing distributional fits for return distribution tails across timescales. Each comparison evaluates the power-law fit versus alternative distributions over the tail range $x \geq x_{\min}$, following the methodology in Alstott, Bullmore, and Plenz [3] and Clauset, Shalizi, and Newman [49]. We choose one distribution over another if the log likelihood ratio favors it and the p -value is below 0.05; otherwise we say the test is inconclusive. Note that each tail is tested separately, thereby giving us 60 results for each timescale (30 symbols, 2 tails per symbol). The log likelihood ratio and p -value for each comparison are provided in Appendix A.9.

tails. Diminishing data size as a function of timescale should be kept in mind for each of these coarse-time findings, however.

2.3.3 GAIN/LOSS ASYMMETRY

Cont details his 3rd stylized fact (gain/loss asymmetry) as prices experiencing larger drawdowns than upward movements. As mentioned in Section 2.1.1, it is not exactly clear what results were being referenced for this fact. Under some interpretations, returns should be expected to show negative skews, with skew measured as:

$$S(\Delta t) = \frac{\langle (r(t, \Delta t) - \langle r(t, \Delta t) \rangle)^3 \rangle}{\sigma(\Delta t)^3}.$$

We do not see this property consistently across symbols for the normalized returns in clock- nor event-time, as shown in Figure 2.6. We see a full range from negative to positive return distribution skewness for each timescale in either clock.

A more literal read of Cont’s description of this fact might imply it applies to tail events, in which case we would expect to see larger losses than we do gains. If the tails of the return distribution behave asymptotically like a power-law, a smaller tail exponent could indicate larger price movements. Note, however, that our estimation of the tail exponents in the previous section (Figure 2.4) did not reveal any consistent or obvious trends to whether the left or right tails had smaller $\hat{\alpha}$ on average. We tested this further by examining the percentage of extreme returns that are negative versus positive for different cutoffs. For a given timescale and quantile q , the cutoff is that quantile of a symbol’s absolute returns in the timescale. The expectation is for most extreme returns to be losses and for this to be more true as the quantile-cutoff

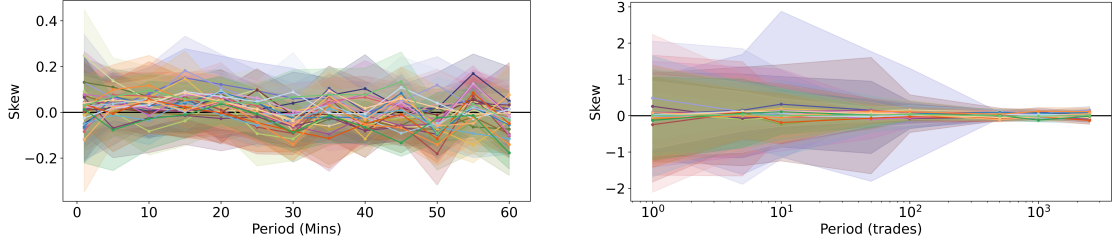


Figure 2.6: Skew of the normalized return distributions in clock- and event-time. Solid lines show the skew for a symbol's full sample of intraday returns. Shaded regions show the 95% confidence interval from bootstrapping.

gets closer to the 100th percentile. We instead found the symbols split between having more extreme gains versus losses in each quantile and timescale¹³. We therefore do not find evidence of a gain/loss asymmetry effect.

2.3.4 VOLATILITY CLUSTERING

Cont's 6th stylized fact expects volatility to cluster in time. We calculate volatility clustering by looking at the autocorrelation of absolute returns. Similar to linear ACF, $C(\tau, \Delta t)$, let

$$C^0(\tau, \Delta t) = \text{corr}(|r(t, \Delta t)|, |r(t + \tau, \Delta t)|).$$

The expectation is for $C^0(\tau, \Delta t)$ to be positive and asymptotically go to zero. More specifically, Cont's 8th stylized fact expects a power-law decay:

$$C^0(\tau, \Delta t) \propto \tau^{-\beta},$$

with $\beta \in [0.2, 0.4]$.

¹³Shown in our supplementary material, Figure A.12.

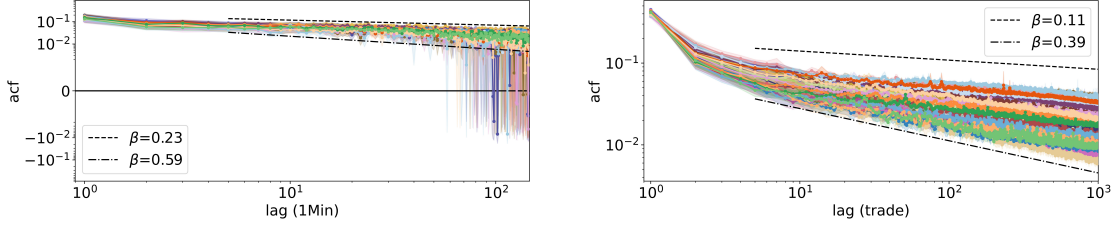


Figure 2.7: Autocorrelation of absolute normalized returns. Solid lines show the absolute autocorrelation for a symbol's full sample of intraday returns. Shaded regions show the 95% confidence interval from bootstrapping. Dashed lines give upper and lower bounds of the power-law exponents β fit to the decay trajectories, starting at lag $\tau = 5$. These lines are offset above and below the empirical results for clarity.

As shown in Figure 2.7, absolute ACF of the normalized 1-minute returns start positive and decay roughly linearly in log-log scale over the more than 100 lags tested. Some symbols have drop offs in the later lags, with negative C^0 for some lags. The normalized trade-time returns show absolute ACF starting above 0.1 with a roughly linear decay in log-log scale past the first few lags. We show guidelines starting at $\tau = 5$ with power-law fits for the smallest and largest β values obtained using log-log regression against the empirical C^0 results. While we find support for the power-law fit over the range tested, we find a wider range of β values than expected (roughly $\beta \in [0.2, 0.6]$ for our one-minute returns and $\beta \in [0.1, 0.4]$ for trade level). Since we only consider intraday returns in this study, we do not look at the correlation of a return from one trading day with a return from the following day. This imposes a natural constraint on how many lags the power-law can hold for under our construction. The ranges where the effect drops off to zero are worth further exploration to better understand the exact shape of decay, the mechanisms behind this shape, and whether alternative descriptions such as a power-law with exponential cutoff might be more appropriate.

2.3.5 INTERMITTENCY

As detailed in Section 2.1.2, the property of intermittency arises from and is closely linked in the literature to the properties of heavy tails and volatility clustering. In particular, large returns occur more frequently than expected under a normal or exponential distribution, and returns of similar magnitude tend to cluster over time. We attempt one additional measure of intermittency through examining the distribution of interarrival times of extreme price moves. Specifically, consider the 99th-percentile largest absolute returns¹⁴ for a given symbol at timescale Δt , with cutoff selected from the symbol's entire 101-day sample. We can count the number of returns $N_{0.99,\Delta t}$ of at least this magnitude in a given period of time, with the degree of variability of $N_{0.99,\Delta t}$ providing a measure of intermittency. We show these extreme returns occur more variably than they would if arising from a Poisson distribution by measuring their Fano factor. The Fano factor is defined as

$$F(\Delta t) = \frac{\sigma_{N_{0.99,\Delta t}}^2}{\langle N_{0.99,\Delta t} \rangle},$$

the ratio of the variance to the mean for the number of extreme returns in a period. We expect this ratio to be greater than one, indicating extreme returns are overdispersed relative to a Poisson distribution. Please note, the Fano factor depends on the dispersion of extreme returns and is therefore sensitive to volatility clustering. We just showed that volatility clustering exhibits long memory and remains positive for well over the MBB block length we use in the other metrics. We thus use a slightly

¹⁴We also tested the Fano factor for 95th-percentile and 97.5th-percentile cutoffs, with similar results. These results are shown in Appendix A.5.

Symbol	1-min Returns	Trade Return
AAPL	1.58 (1.24–1.96)	56.15 (48.47–60.98)
AXP	1.61 (1.35–1.94)	45.69 (41.08–50.37)
BA	1.92 (1.47–2.47)	56.57 (51.52–61.65)
CAT	1.76 (1.39–2.27)	53.72 (48.78–58.57)
CSCO	1.48 (1.22–1.76)	44.58 (36.19–49.50)
CVX	1.33 (1.18–1.48)	49.51 (41.92–55.25)
DIS	1.52 (1.27–1.78)	54.90 (49.02–60.48)
DWDP	1.43 (1.22–1.62)	49.58 (33.58–65.70)
GS	1.49 (1.26–1.74)	45.46 (41.36–49.37)
HD	1.51 (1.31–1.74)	56.43 (49.60–61.52)
IBM	1.59 (1.35–1.86)	54.77 (46.05–62.94)
INTC	1.37 (1.22–1.52)	47.86 (38.31–53.99)
JNJ	1.75 (1.41–2.21)	72.08 (62.82–79.99)
JPM	1.48 (1.29–1.68)	44.75 (38.85–50.92)
KO	1.46 (1.28–1.66)	40.52 (33.29–45.28)
MCD	1.49 (1.23–1.83)	58.78 (53.95–63.43)
MMM	1.60 (1.36–1.84)	48.42 (44.27–52.31)
MRK	1.35 (1.21–1.49)	32.92 (28.24–38.21)
MSFT	1.61 (1.35–1.87)	53.62 (45.23–61.16)
NKE	1.45 (1.14–1.82)	38.50 (33.80–43.66)
PFE	1.28 (1.12–1.46)	34.85 (29.85–38.19)
PG	1.46 (1.23–1.73)	48.17 (43.14–52.78)
TRV	1.46 (1.19–1.79)	42.27 (38.19–46.16)
UNH	1.46 (1.30–1.63)	56.17 (49.76–62.28)
UTX	1.72 (1.36–2.08)	56.86 (51.74–61.56)
V	1.58 (1.28–1.94)	54.63 (48.77–60.55)
VZ	1.35 (1.22–1.47)	42.74 (32.13–50.55)
WBA	1.38 (1.21–1.53)	41.90 (37.34–46.59)
WMT	1.58 (1.28–1.93)	48.18 (42.81–52.96)
XOM	1.35 (1.16–1.56)	44.79 (28.57–57.67)

Table 2.3: Fano factor of the number of extreme (99^{th} -quantile) returns that arrive within a coarse period of time. The coarse period for normalized 1-minute returns is 30-minutes, and the coarse period for trade-level returns is 1000 trades. Values in parentheses give the 95% bootstrap confidence interval for the metric.

different bootstrap method to calculate 95% bootstrap confidence intervals for the Fano factor. We use each trading day as a block, selecting (with replacement) 101 trading days for each bootstrap sample. We construct 1,000 bootstrap samples in this way to estimate a 95% confidence interval for the metric.

The Fano factor and confidence intervals are summarized in Table 2.3. We observe that the Fano factor for extreme trade-level returns in blocks of 1000 trades is well above one, indicating strong overdispersion relative to a Poisson benchmark. For extreme 1-minute returns in 30-minute windows, the Fano factor is closer to (but still above) one, suggesting overdispersion with somewhat less ‘burstiness.’ Part of this difference may be due to smaller sample sizes at the 1-minute scale—there are only 390 observations per day, implying roughly 3.9 extreme returns on average at the 99th-percentile threshold. In both trade-time and clock-time data, the distributions of intraday interarrival times for extreme returns are heavy-tailed, with excess kurtosis of approximately 2–6 for 1-minute intervals and even larger values for trade-level events¹⁵. Together with prior evidence of heavy tails and volatility clustering, these findings point to clear signs of intermittency in both clock-time and event-time.

2.3.6 LEVERAGE EFFECT

Cont’s 9th stylized fact asserts that volatility is negatively correlated with the returns for an asset. We measure this ‘leverage effect’ as Cont [54] laid out, drawing from the results of Bouchaud, Matacz, and Potters [27] and Pagan [143]. This approach looks simply at the correlation between returns and lagged volatility, with volatility

¹⁵Kurtosis values for the distribution of extreme intraday interarrival times are shown in our supplementary material, Table A.1.

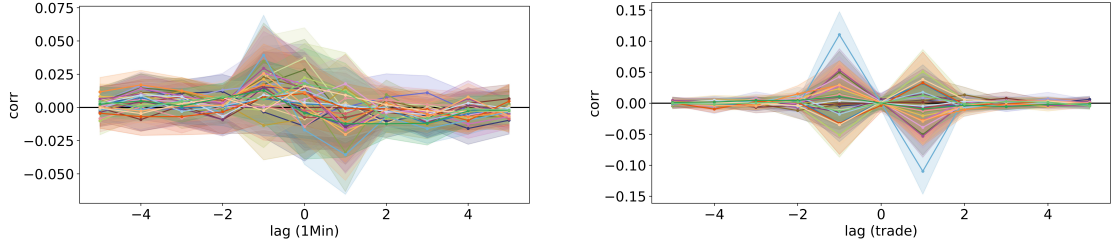


Figure 2.8: Leverage effect $L(\tau, \Delta t) = \text{corr}(r(t, \Delta t), |r(t + \tau, \Delta t)|)$ for the normalized returns. Solid lines show the correlation for a symbol's full sample of intraday returns. Shaded regions show the 95% confidence interval from bootstrapping.

measured as the absolute returns¹⁶:

$$L(\tau, \Delta t) = \text{corr}(r(t, \Delta t), |r(t + \tau, \Delta t)|).$$

The expectation is for $L(\tau) < 0$ when $\tau = 1$ and for $L(\tau) < L(-\tau)$ when $\tau > 0$.

We find no clear trend to the correlation values across symbols. There are varying strengths and signs to the correlations at each lag, suggesting they might arise from specific variation in the path of a given symbol's price over our observation period. For some symbols, there is an interesting symmetry where $L(-1) \approx -L(1)$, particularly for the normalized trade-level results. This also goes against the descriptions from Cont [54] and Bouchaud, Matacz, and Potters [27], who described the relationship between returns and negatively lagged volatility as being largely insignificant. Overall, we do not see the expected direction of the leverage effect relationship, nor do we see any clear trend in this relationship across the symbols in either timescale.

¹⁶We also looked at the squared returns for the volatility, as used by Bouchaud, Matacz, and Potters [27]. There was no material effect to our results, as shown in our supplemental material (Figure A.15), and we therefore use the absolute returns to be consistent with the rest of our analysis.

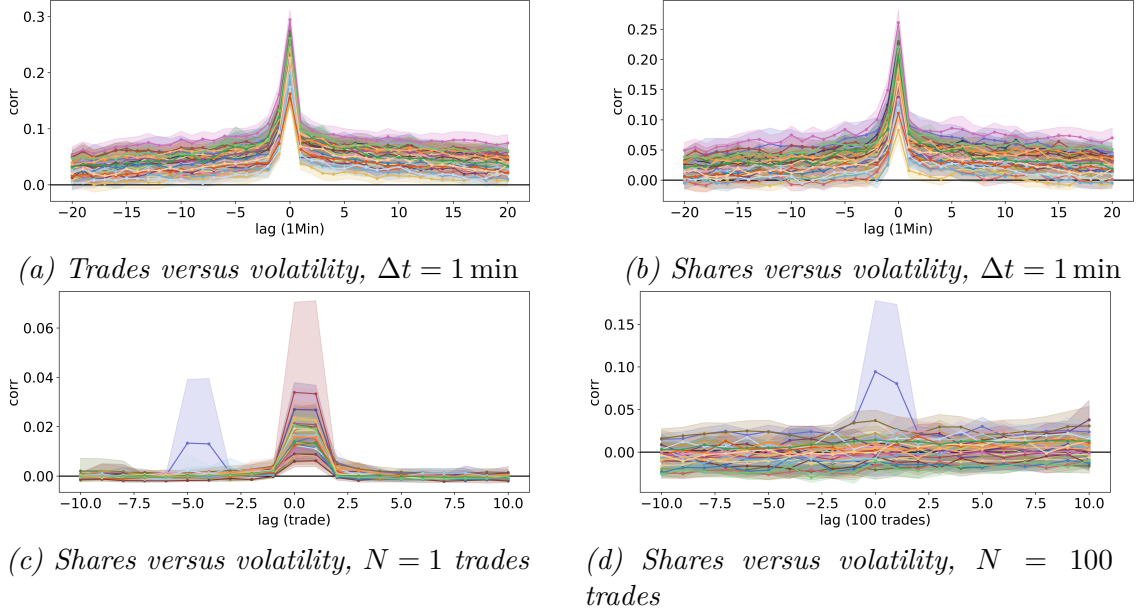


Figure 2.9: Volume versus lagged volatility: $V(\tau, \Delta t) = \text{corr}(\text{volume}(t, \Delta t), |r(t + \tau, \Delta t)|)$. Solid lines show the correlation for a symbol's full sample of intraday returns and volume. Shaded regions show the 95% confidence interval from bootstrapping.

2.3.7 VOLUME/VOLATILITY CORRELATION

The relationship between volume and volatility is Cont's 10th stylized fact and asserts a positive correlation between volume and volatility. Let $\text{volume}(t, \Delta t)$ denote the volume for period $[t - \Delta t, t]$, where volume is either the number of trades or shares in that period. We examine the correlation between volume and lagged volatility, with volatility again measured as the absolute returns:

$$V(\tau, \Delta t) = \text{corr}(\text{volume}(t, \Delta t), |r(t + \tau, \Delta t)|).$$

Volume has an intraday u-shaped activity pattern in clock-time similar to the volatility pattern that we correct for when normalizing our clock-time returns (Section 2.2).

For the clock-time metric, we therefore similarly normalize for this pattern by dividing $volume(t, \Delta t)$ by the average volume in that period across the 101-day sample for a given symbol¹⁷. We look at both trades and shares as possible volume metrics in clock-time, with similar results (Figures 2.9a and 2.9b). We see a strong contemporaneous positive correlation between volume traded and volatility for each symbol. This relationship is slightly stronger on average between trade volume and volatility than for share volume. The effect decays quickly for both volume measures versus past or future volatility, with some symbols going to approximately zero in fewer than 10 lags. Other symbols show positive correlation for each of the 20 lags we tested in either direction.

In event-time, the number of trades in a period is fixed, but the volume of shares can vary across different sets of N trades. We first tested the relationship between share volume and volatility for the normalized trade-level returns. There is a slight (<0.05) positive correlation across the symbols for $\tau \in \{0, 1\}$, and the bootstrap confidence intervals stay above zero for these lags as well. However, the effect is so weak, we also looked at $N = 100$ in case the effects of varying share volume need multiple trades to accumulate. Instead, we find weak correlation of varying sign at this level of aggregation. Taken together, these results do not show a clear correlation between share volume and volatility in event-time. The presence of volume-volatility correlation in clock-time but not event-time gives some explanation towards differences in how trade-level and clock-time returns behave¹⁸.

¹⁷The raw 1-minute returns show stronger correlation with the raw volume (Appendix A.1) than the normalized relationship shown here. The correlation stays positive for more than 50 lags in both directions for the raw returns and volume as well. Normalization by the daily activity patterns thus corrects for much of this relationship.

¹⁸Indeed, we see a slow decay in the autocorrelation of trade volume in clock-time, similar to the volatility clustering discussed in Facts #6 and 8. This is shown in Appendix A.3.

2.3.8 ASYMMETRY IN TIMESCALES

The final stylized fact (Fact #11) examines the asymmetry of the flow of information across timescales. For our base timescale Δt of 1-minute, we consider the coarse-grained timescale $\Delta T = 30$ min. In event-time, we use a base timescale of trade-time and $N = 1000$ as our coarse timescale ΔT . We calculate

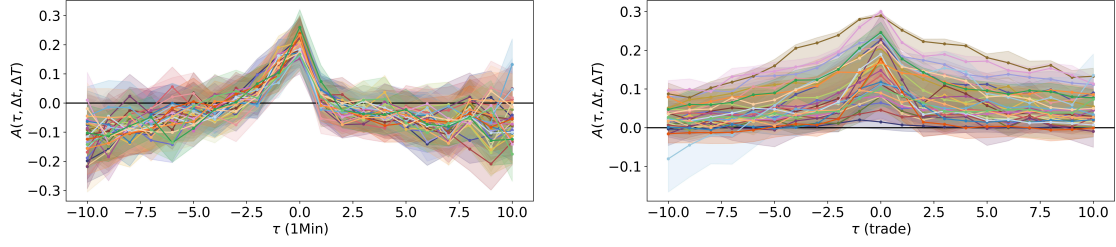
$$A(\tau, \Delta t, \Delta T) = \text{corr} \left(\overline{|r(t \in T, \Delta t)|}, |r(T + \tau, \Delta T)| \right),$$

as done by Müller et al. [138]. The fine- and coarse-grained returns are each normalized at their respective timescales prior to calculation of $A(\tau, \Delta t, \Delta T)$. We measure the asymmetry by differencing the correlation at the corresponding positive and negative lags τ :

$$D(\tau, \Delta t, \Delta T) = A(\tau, \Delta t, \Delta T) - A(-\tau, \Delta t, \Delta T).$$

The expectation is for negative lags to have a larger correlation than the corresponding positive lags for at least a few steps.

We plot $A(\tau, \Delta t, \Delta T)$ in Figure 2.10 and $D(\tau, \Delta t, \Delta T)$ in Figure 2.11. First, note that there is clear contemporaneous correlation between the fine and coarse volatility. This correlation stays positive for each of the symbols at $\tau = -1$, while some symbols drop below zero for $\tau = 1$. The correlation is weaker and has varying sign across the symbols for each lag $|\tau| > 1$ in the clock-time results. In event-time, at least one symbol has bootstrap confidence intervals that include zero for $\tau = 0$. The nonzero correlations that are seen decrease with τ in either direction, similar to the clock-time results. These results are already contrary to the expected effect, which would expect



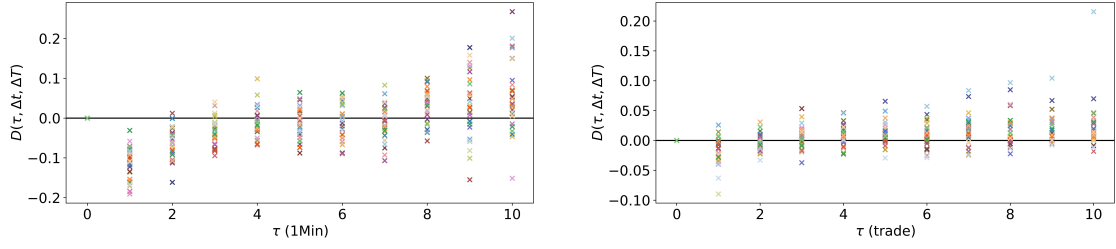
(a) 1-minute versus lagged 30-minute volatility for the normalized clock-time returns.

(b) Trade versus lagged 1000-trade volatility for the normalized event-time returns.

Figure 2.10: Average fine versus coarse volatility, $A(\tau, \Delta t, \Delta T) = \text{corr}\left(\overline{|r(t \in T, \Delta t)|}, |r(T + \tau, \Delta T)|\right)$. Solid lines show the correlation for a symbol's full sample of intraday returns. Shaded regions show the 95% confidence interval from bootstrapping.

a strong correlation between coarse- and fine-grained volatility [138].

We find little evidence of the expected asymmetry when we look at $D(\tau, \Delta t, \Delta T)$ (Figure 2.11), which we would expect to be negative for at least a few lags. Each of the symbols have $D(\tau, \Delta t, \Delta T) < 0$ at the first-lag difference in clock-time, but there is varying strength to this difference. The smallest difference is less than 0.05 in magnitude. Beyond the first lag, symbols are more evenly split, and we know from Figure 2.10 that some of those differences are between values of A that are roughly zero or even negative. For contrast, Müller et al. [138] found the correlation between fine- and coarse-volatility stayed positive for more than 20 lags in either direction, the first-lag difference was ≈ 0.1 , and the differences stayed negative for more than 10 lags. It is possible the asymmetry effect is more plausible when considered for symbols on average, at least in clock-time. We discuss this possibility more in the next section. However, aside from a slight negative difference at the first-lag in clock-time, we do not find a consistent negative difference for individual securities to indicate a clear asymmetry of volatility prediction across timescales.



(a) 1-minute versus lagged 30-minute volatility for the normalized clock-time returns.

(b) Trade versus lagged 1000-trade volatility for the normalized event-time returns.

Figure 2.11: Asymmetry effect—The difference between positive and negative lags for the correlation between the average fine volatility and lagged coarse volatility.

2.4 DISCUSSION

Cont’s original set of stylized facts [54] emerged from a synthesis of empirical studies, each study focused on a market which existed prior to 2001. As demonstrated elsewhere, the technological arms race and resulting market fragmentation in the intervening decades since Cont’s study fundamentally changed the dynamics of the U.S. stock market [106, 173, 184]. Motivated by these market changes to revisit Cont’s original study, we find strong evidence for eight of Cont’s 11 stylized facts when looking at intraday returns for each of the Dow 30 stocks in our sample.

As summarized in Table 2.1, how we construct our timeseries of returns matters in how (or whether) the facts manifested. We find positive correlation between volume and volatility (Fact #10) in clock-time but not in general for the event-time returns. Meanwhile, aggregational Gaussianity (Fact #4) is more clearly expressed after normalizing the returns by activity level. These nuances have implications for our understanding of stock markets and how we should evaluate models thereof. Most agent-based models of markets are evaluated on their simulated price series, which

are frequently more analogous to event-time in the empirical market than to clock-time [41]. The developer of such a model can take confidence in using at least seven of Cont’s stylized facts to evaluate their model’s relevance to the empirical market, while caution should be taken if expecting a strong volume/volatility relationship depending on how time is being evaluated.

The measures we examine for Facts #3, #9, and #11 do not provide evidence of gain/loss asymmetry, the leverage effect, or asymmetry in timescales, respectively, being consistently present for intraday returns of individual stocks. Instead, the measures vary in sign and strength depending on the symbol and timescale. Alternative approaches looking at longer timescales (e.g., daily) or grouping stocks together might yield different results for these facts. An example is the analysis done by Chicheportiche and Bouchaud [47], who looked at daily data from 280 stocks from 2000–2009. They standardized the returns such that they could measure the leverage effect and asymmetry in timescales for all stocks together, finding evidence of both effects. Blanc, Donier, and Bouchaud [22] took a similar approach using 5-minute returns of 133 stocks from 2000–2009. They found evidence of asymmetry in timescales but noted that the leverage effect is insignificant for intraday returns. Another approach taken in the literature is to look at the motion of multiple stocks together (such as an index or the average price over multiple stocks). Bouchaud, Matacz, and Potters [27] found the average leverage effect for daily index prices was stronger than for daily stock prices. Studies measuring gain/loss asymmetry through investment horizons found the effect to appear for indices but not for individual stocks [104, 105, 166]. Future research could therefore examine the full set of Cont’s stylized facts for groups of stocks collectively or using daily prices.

Our results provide an improved understanding of how Cont’s stylized facts should be interpreted in modern markets. In particular, eight of the stylized facts appear to be well supported for intraday returns of individual stocks, while the remaining three were not present for the measures examined. Cont’s stylized facts are not exhaustive in capturing characteristics of financial markets. Examining the commonality of other features of financial markets such as daily volatility patterns [4, 197], extreme price events [106, 127], and order book characteristics [89] can help widen our understanding of stylized facts and provide additional benchmarks for market models to hit. A robust set of stylized facts serves at least two distinct communities. For the community of financial regulators, the set of stylized facts provides guideposts against which to assess both the intended and unintended impacts of regulatory reform. For the scientific community, the set of stylized facts provides the guideposts for the design, development, testing, and calibration of the next generation of market models.

CHAPTER 3

TESTING REPLICATION FOR AN AGENT-BASED MODEL OF MARKET FRAGMENTATION AND LATENCY ARBITRAGE

ABSTRACT

This study strengthens the foundations of multi-venue market modeling by attempting an independent replication of Wah and Wellman’s 2016 model of latency arbitrage in a fragmented market. We find that faithful replication is hindered by missing implementation details in the original paper and limited quantitative reporting. We demonstrate that increasing the number of simulation runs beyond the original design allows for the creation of bootstrap confidence intervals to support rigorous tests of quantitative alignment, compensating for lacking distributional information (e.g., variance). We also demonstrate that increased complexity across the modeled sce-

narios corresponds with increased difficulty aligning to the original results. We draw on a codebase released by the original authors in connection with a later paper to recover additional implementation details; however, we reject quantitative alignment between that codebase and the published results. Combining information from the paper and the released code, we achieve relational equivalence for most metrics but reject quantitative alignment for model settings where latency is nonzero. We show that many of the qualitative takeaways from the original paper on the effects of market fragmentation and latency arbitrage are sensitive to the specifics of a ‘greedy strategy’ extension given to the zero-intelligence (ZI) trader agents. Under an alternative interpretation of this strategy, we find that market fragmentation decreases execution times in all experiments and increases trader welfare in most experiments. Finally, to facilitate future replication, critique, and extension, we provide an ODD (Overview, Design concepts, Details) protocol for our implementations of the model.

3.1 INTRODUCTION

The main regulator of U.S. financial markets, the Securities and Exchange Commission (SEC), has recently stated that existing market-simulation frameworks are not sufficiently mature to guide its policymaking [180]. They point to a lack of existing frameworks for studying the structural considerations they are trying to address, and they warn:

Challenges associated with simulations include model accuracy and complexity, data quality, computational limitations, uncertainty and sensitivity, validation and verification, scalability, and the challenges associ-

ated with applying simulations to human behavior which is inherently unpredictable, and the challenges associated with modeling dynamic and evolving systems like financial markets. (p. 328)

Essential steps towards addressing these concerns are the development of computational models that better capture the complexity of real-world markets and independent replication to establish the rigor of insights from such models.

Towards these ends, we present here the results from our attempt to replicate Wah and Wellman’s influential agent-based model (ABM) of latency arbitrage in a fragmented market [189, 190]. To our knowledge, their simulation framework was the first to include multiple exchanges and communication latency—defining features of the modern U.S. stock market, which comprises 18 exchanges as of July 2026 [40]. Under this framework, Wah and Wellman report that market fragmentation and latency arbitrage negatively impact market liquidity and efficient allocation of value to traders. The model is an important building block for empirically-relevant market models [66, 175, 183], and it has been cited in a number of studies and policy briefs discussing alternative market designs (e.g., Baldauf and Mollner [18], Brolley and Cimon [31], Budish, Cramton, and Shim [32], Chicago Stock Exchange, Inc. [46], and Securities and Exchange Commission [158, 157]).

Our motivation was to use a successful replication of the model as a rigorous baseline for planned extensions targeting specific aspects of fragmentation in the modern U.S. stock market. We found, however, that full replication is hampered by undocumented implementation choices and sparse statistical reporting. Our main results are the following:

1. Increasing the number of simulation runs an order of magnitude above the

original experimental design helps rigorously test quantitative alignment, compensating for lacking distributional information (e.g., variance).

2. Increased complexity across the modeled scenarios corresponds with increased difficulty aligning to the original results.
3. Qualitative alignment (i.e., relational equivalence) across liquidity and efficiency metrics can be achieved only after consulting with a codebase released by the original authors connected to a later study.
4. The top-level takeaways from the model are sensitive to the specifics of a ‘greedy strategy’ extension given to the zero-intelligence (ZI) traders; under an alternative implementation, we find that market fragmentation overall decreases execution times and increases trader welfare.

Taken together, these findings highlight the value of independent replication in the modeling process while underscoring the need for more computational study of market fragmentation and improved reporting of market model assumptions and results.

The rest of the paper is organized as follows. In the next section, we give background on the role of replication in the agent-based modeling and agent-based market modeling literature. We then provide an overview of Wah and Wellman’s model. We introduce our quantitative alignment methodology and use this methodology to test three implementations of the model against the canonical results. We then compare the qualitative behavior of these implementations to the original results before summarizing our results and concluding. We also provide an ODD (Overview, Design concepts, Details) protocol [90, 91] for the model in our appendix, to facilitate future replication, critique, and extension.

3.2 BACKGROUND

Independent replication is a cornerstone of scientific practice. It verifies whether the methods described by the original study are sufficient to produce the results claimed and allows for the extension of these methods for further study [11, 185]. While still relatively rare in the agent-based modeling literature [11, 97], its value to the practice has been demonstrated numerous times. Unsuccessful replication highlights where the documented description is not sufficient to reproduce the original findings [195], ideally leading to clarification and further analysis [122, 194]. Successful replication efforts often identify sensitivity of the original findings to certain implementation details, thereby refining our understanding of the model’s assumptions and scope [67, 129, 171]. The replicated model can be checked for robustness to additional experiments, analysis, and extensions [34, 94, 171]. More broadly, the practices of replication, extension, and alignment allow for the cumulative development of more complex models built on robust foundations [11, 13].

In their seminal study on aligning computational models, Axtell et al. [13] introduce a comparison framework with three different levels of ‘equivalence’ between models. The strongest level of equivalence is numerical identity, where both models produce identical outputs given the same settings. This will generally not be feasible for models with stochastic components unless the replication shares roughly the same code (including pseudorandom number generators and seeds). The next level is distributional equivalence, where the distributions of the two sets of results cannot be distinguished statistically from one another. Axtell et al. highlight that distributional detail of the reported measurements is needed in order for equivalence to be

statistically tested in a replication effort. The weakest level is relational equivalence, under which the models produce the same qualitative relationships between inputs and outputs.

Axtell et al. [13] also note that a precise, detailed description of the model is vital to replication. Major effort has been undertaken to encourage rigorous and standardized model descriptions to aid in this process [52, 67, 90, 186]. A prevailing format is the Overview, Design concepts and Details (ODD) protocol introduced and refined by Grimm et al. [90, 91]. Szangolies et al. [169] propose a standardized visual ODD (vODD) format to provide a quick overview of the model to supplement the more detailed written description. Grimm et al. [93] recently demonstrated how the ODD protocol can be used to aid in the replication process, ambitiously aiming to replicate 18 models and succeeding to at least some degree for 15 of them.

3.2.1 REPLICATION OF AGENT-BASED MARKET

MODELS

Independent replication is particularly rare for agent-based market models, and published claims have generally not exceeded relational equivalence. Preis et al. [149] assert that the Farmer et al. ‘zero-intelligence’ model (ZIM) [61, 167] is a special case of their more complex market model, claiming reproduction of the ZIM results without providing evidence in the paper. Tivnan et al. [172] integrate the ZIM market environment with a richer trader ecology from Ghoulmie, Cont, and Nadal [85]. Tivnan et al. note that they lacked the code and quantitative results from the original models and therefore could not test distributional equivalence. They claim success

in establishing relational equivalence but do not display the results of that exercise in the paper due to space constraints. Bookstaber, Foley, and Tivnan [24] assert replication and extension of the Preis et al. model, showing their base replication produces similar stylized facts to the Preis et al. model but not providing evidence of equivalence in the paper. Duffin and Cartlidge [66] replicate and extend Wah and Wellman’s 2013 two-market model [190], demonstrating relational equivalence for a single metric. Similar to Tivnan et al., they could not test distributional equivalence due to lack of quantitative results from the original paper.

Another route in developing cumulative progress in the agent-based market modeling space has been through sharing model code as open-source. An early success in this vein was the Artificial Stock Market (ASM) [113, 144], which was built in the 1990s by researchers at the Santa Fe Institute, released online by the original team, and reimplemented in different coding languages by others [68, 107, 146]. Multiple versions allowed for comparison between implementations, including analysis by Polhill et al. [146, 147] on how differences in floating point arithmetic can produce divergent behavior. A number of open-source financial ABM platforms have been released in recent years [20, 36, 50, 62, 126]. This allows for the development of computational experiments upon common frameworks, but it does not mean that subsequent works are necessarily replicating and verifying the results of studies that came before. Also, while multiple of these open-source platforms tout the ability to implement multiple exchanges in a market [36, 126], they have not generally been used to model such a scenario.

3.3 THE MODEL

The Wah and Wellman model [189, 190] examines the effects of fragmentation, latency, and latency arbitrage in a financial market. The model features a single asset traded by investor (‘background’) agents in a market with one or more exchanges. A consolidated market with a single exchange is simulated first, after which an additional exchange, communication latency, and an arbitrage agent are incrementally added to examine their respective effects. The specific type of arbitrage examined is an idealized form of latency arbitrage, in which a trader agent with faster access to information in the market takes advantage of price discrepancies to earn a risk-free profit. The slower background traders get cross-market information at a lag from the Security Information Processor (SIP), a data consolidator for the market. The hypothesis is that fragmentation and unequal access to cross-market information impact the efficient allocation of value to investors and the liquidity of the market overall.

Wah and Wellman’s original 2013 model [190] features zero-intelligence (ZI) traders that each act exactly once per simulation according to a fixed, uniform trading strategy based on Gode and Sunder [86]. The following features were varied: the number of exchanges, the amount of SIP latency, and the absence or presence of a latency arbitrage (LA) agent. All other parameters were fixed across the experiments. Wah and Wellman [190] ran 200 simulations for each experiment and reported that latency arbitrage in fragmented markets reduces total surplus and has mixed effects on liquidity. Duffin and Cartlidge [66] replicated and extended the 2013 model in 2018, demonstrating relational equivalence for one metric (trader surplus) but not numerical agreement. They show sensitivity of the model results to the ZI shading

parameter and obtain qualitatively different results when the background traders are updated to use ‘zero-intelligence plus’ (ZIP) [51] strategies.

Wah and Wellman substantially expanded the model in 2016 [189]. In the updated version, ZI traders participate multiple times per simulation, shade orders using more sophisticated surplus-seeking logic, and have heterogeneous strategy parameters. For each experiment, Wah and Wellman [189] compute an equilibrium distribution of ZI trader strategies, where each agent aims to maximize its expected payoff (surplus). They ran the model for three sets of environment parameters, varying market configurations for each environment and running 50,000 simulations for each experiment. They found latency arbitrage reduces surplus and liquidity in the updated model, while fragmentation has mixed effects.

We focus on replicating the 2016 version [189] of the model here. This decision is based on the 2016 study’s more advanced ZI trader behavior and richer experimental design relative to the original 2013 study [190]. The 2013 study features only a single set of environment parameters and presents results only graphically, without quantitative summary measures. Duffin and Cartlidge [66] similarly only report results visually rather than quantitatively. Wah and Wellman [189] ran the model for three environments and reported mean surplus values for each experiment, providing more information to benchmark our replication against. The 2016 study’s results are also based on 50,000 simulations for each experiment, orders of magnitude more than the 200 simulations run per experiment in the 2013 study. Finally, the extended ZI agent strategy and mixture of strategy parameters used in the 2016 model differ from the basic ZI agents critiqued by Duffin and Cartlidge [66], making it unclear how their critique applies to the updated model.

We refer to the 2016 study [189] simply as WW going forward. We detail the high-level logic of the WW model in the following sections. We follow an abbreviated version of the ODD (Overview, Design concepts, Details) protocol [90, 91] here, highlighting ambiguities from the original text and corresponding design choices we made. We also provide a high-level visualization of the model following the vODD format [169] in Fig. 3.1. We give a more complete accounting of the model following the ODD protocol in our appendix.

3.3.1 ENTITIES, STATE VARIABLES, AND SCALES

Entities, state variables, and scales are summarized in Table 3.1, Table 3.2, and Table 3.3, respectively.

Entities	Scheduler: schedules and kicks off events in the simulation Exchange: market operating a continuous double auction (CDA) Security: fungible asset traded on each exchange Security Information Processor (SIP): data consolidator for the market; maintains and publishes the National Best Bid and Offer (NBBO) Traders: Zero Intelligence (ZI) background traders Latency Arbitrage (LA) trader
-----------------	--

Table 3.1: ODD protocol entities for WW.

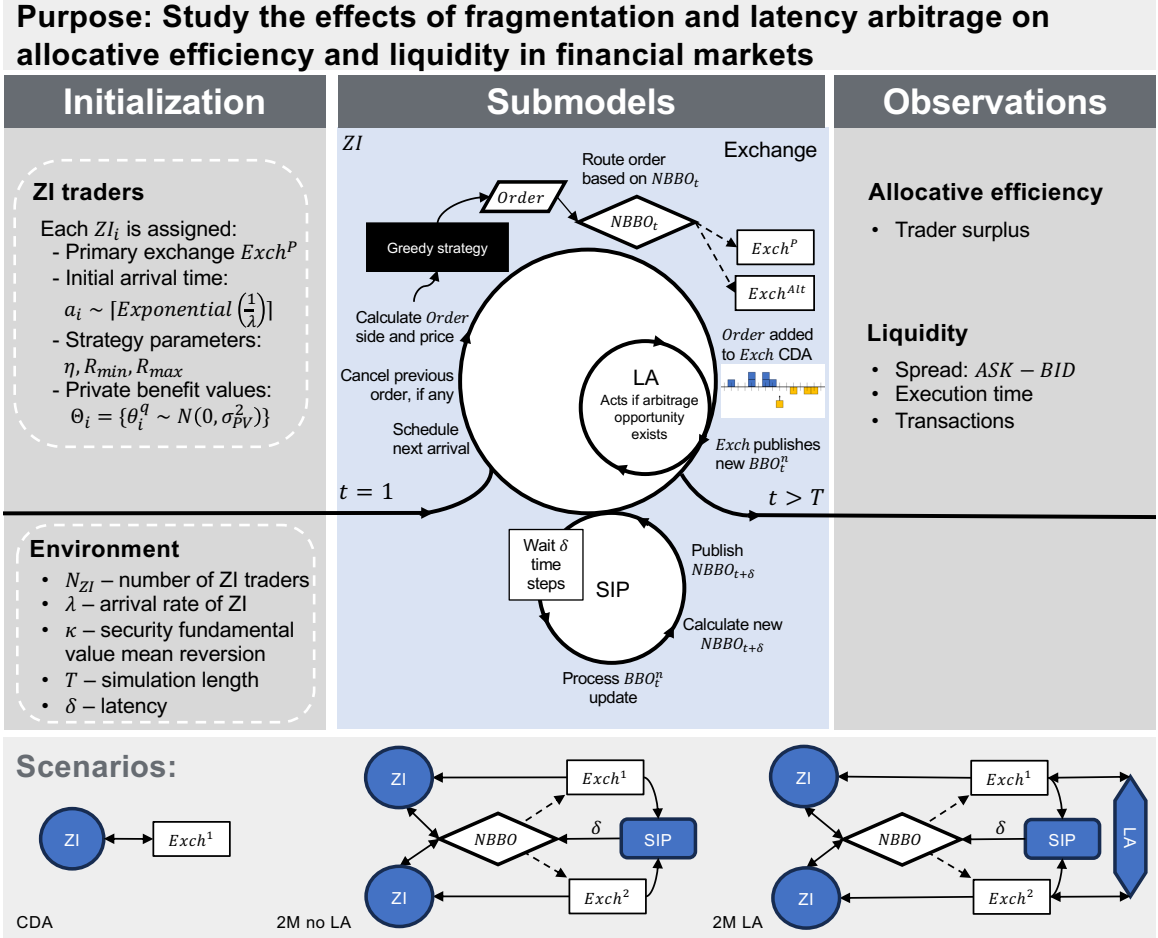


Figure 3.1: Visualization of WW following the vODD format [169]. The model examines the effects of fragmentation and latency arbitrage in financial markets by varying the number of exchanges, SIP latency, and presence of a latency arbitrage (LA) trader in the market. At initialization, a market is set up according to the experimental settings. Then, N_{ZI} zero-intelligence (ZI) traders are created. Each is assigned a primary exchange, private strategy features, and an initial arrival time. ZI arrivals drive the events of the model. Upon arrival, a ZI trader will submit an order to an exchange, resulting in BBO feed updates, possible arbitrage action by the LA agent (if it is present), and updates to the SIP NBBO feed at some latency $\delta \geq 0$. Metrics of market efficiency and liquidity are averaged over many runs of the model for each experiment and then compared across the different scenarios.

State variables	Scheduler Current time t List of scheduled events (ZI arrivals and SIP updates) Exchange Limit order book (LOB): all resting orders with prices, quantities, and timestamps Best bid and offer (BBO): highest buy price and lowest sell price on the exchange BBO subscribers SIP Exchange BBO(s): Current BBO feeds from the exchange(s) National Best Bid and Offer (NBBO) NBBO subscribers Security Fundamental value r_t ZI trader Valuation vector Θ_i : marginal private benefits for buying and selling the asset Strategy parameters $\{R_{\min}, R_{\max}, \eta\}$ Primary exchange Primary exchange BBO Current holdings: quantity q_i the trader is long or short Outstanding orders Profits from trading LA trader (if present) Exchange BBOs Profits from trading
------------------------	--

Table 3.2: ODD protocol state variables for WW.

Scales	T : Simulation length N_{type} : Number of background traders of a given type $N_{exch} \in \{1, 2\}$: Number of exchanges $\delta \geq 0$: SIP latency $N_{LA} \in \{0, 1\}$: Presence or absence of the LA trader λ : ZI arrival rate q_{max} : maximum shares ZI trader can be long or short the asset
---------------	--

Table 3.3: ODD protocol scales for WW.

3.3.2 PROCESS OVERVIEW AND SCHEDULING

Time is discrete, with steps $t \in \{1, \dots, T\}$. A scheduler maintains a queue of events, ordered by time and then by insertion order (first-in, first-out). Only two types of events go through the scheduler queue: ZI trader arrivals and SIP NBBO update events (scheduled with latency δ after a quote change). All other actions in the model are triggered immediately (with zero modeled latency) as consequences of scheduled events. Non-scheduled actions include order submissions and cancellations, exchange matching and trade execution, exchange BBO updates, LA trader responses, and notifications of trades.

3.3.3 EXCHANGES

The market can be configured with one or two exchanges. Each exchange in the model operates a continuous double auction (CDA)¹ for the asset. A CDA is a two-sided market where traders submit orders to buy and sell the asset. A high-

¹Wah and Wellman [189, 190] also examined an alternative clearing mechanism, periodic call auctions, as a possible remedy to latency arbitrage. All national exchanges in the U.S. stock market currently use CDAs, however, and we leave the periodic call auction models out of scope for now.

level visualization of a CDA is given in Fig. 3.2. The market is ‘continuous’ in the sense that orders and trades occur as soon as they are received and processed by the exchange’s matching engine². A limit order³ to buy (sell) the stock specifies a maximum (minimum) price the agent is willing to trade at and a quantity to buy (sell). In the model, all orders are for a single unit of the stock. Each exchange maintains a limit order book (LOB), constructed with the following elements:

1. Orders: map from order IDs to their respective orders
2. Bids: map from price to the queue of buy order IDs at that price
3. Asks: map from price to the queue of sell order IDs at that price

The exchange publishes a BBO feed to its subscribers, containing the best bid and offer prices in the LOB at that moment. LOBs are initialized with no bids or asks at the start. The BBO has null prices for its best available bid and ask whenever the respective side of the LOB is empty (including at initialization).

When an exchange receives an add message for order O_i with limit price p , it does the following. First, it checks whether the added price can be matched against existing orders in the book. More explicitly, if order O_i is a buy order, the exchange checks whether there is a resting sell order O_j with price $q \leq p$. Similarly, if O_i is a sell order, the exchange checks whether there is a resting buy order O_j with price $q \geq p$. In either of these cases, the orders are matched at price q , resulting in a transaction. If multiple resting orders exist on the LOB that could match with the

²There is a time resolution below which the time of two different messages cannot be differentiated, but for modern exchanges this is on the order of microseconds or even nanoseconds [8, 183].

³A market order is an alternative type of order which executes immediately at the best available price. All agents in the model exclusively submit limit orders, however, and so market orders are out of scope.

incoming order, the oldest order is given priority (i.e., price-time priority). In this model, all orders are for a single quantity of the asset, and so each trade is always for a single unit. Receipts are sent to the traders associated with O_i and O_j , and O_j is removed from the LOB. If O_i instead does *not* immediately match, it is added to the LOB. A trader can request to withdraw a resting order from the LOB. Upon receiving a withdraw request, the exchange removes the corresponding order from the LOB.

After any change to the exchange's LOB due to a withdraw, add, or trade, the exchange publishes its updated best bid and ask prices to its subscribers. This is published regardless of whether the best bid or ask prices were changed by the LOB update. The exchange loops over its subscribers in the order they subscribed to the exchange's BBO feed, directly calling the subscriber's `update_bbo()` function.

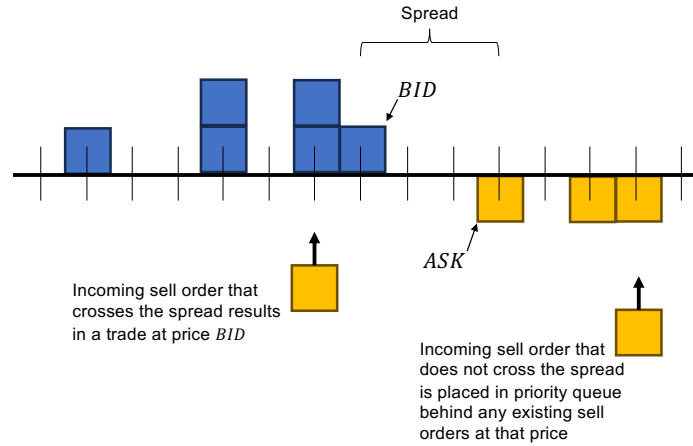


Figure 3.2: High-level logic for a simple continuous double auction (CDA) market. Buy and sell limit orders are shown as blue and yellow boxes, respectively. Example logic for incoming sell limit orders is shown, with one order resulting in a trade and the other resulting in a new resting order, queued behind an existing sell order at the same price. Note that in this example the spread ($ASK - BID$) is of width three and will increase to four after the trade. Logic for incoming buy limit orders is analogous. Each of the 18 national exchanges in the U.S. stock market currently operates a CDA. The Wah and Wellman [189, 190] model is one of only a few agent-based market models to feature a fragmented market with multiple exchanges operating CDAs.

3.3.4 SECURITY INFORMATION PROCESSOR (SIP)

The SIP consolidates information for the market. It subscribes to the BBO feed from each exchange, publishing the following information in its National Best Bid and Offer (NBBO) feed: the overall highest bid, the exchange holding the highest bid, the overall lowest ask, and the exchange holding the lowest ask. Any time an exchange publishes a BBO update, the SIP schedules an NBBO update event with the new information to occur at $t + \delta$, where t is the time the BBO was received and δ is the SIP latency parameter. When the scheduled NBBO update occurs, the updated feed is instantaneously communicated to all SIP subscribers. We assume that when latency $\delta = 0$ the SIP should update and publish its NBBO without going through the scheduler.

3.3.5 SECURITY

The security being traded in the market is mostly abstract, except it possesses an evolving ‘common fundamental value’ known by the traders. The fundamental time series follows a mean-reverting random walk:

$$r_t = \max\{0, \kappa r_{\text{mean}} + (1 - \kappa)r_{t-1} + u_t\}, \quad (3.1)$$

where parameter $\kappa \in [0, 1]$ specifies the strength of reversion to the mean r_{mean} , and $u_t \sim N(0, \sigma_{\text{shock}}^2)$ is a Gaussian random shock at each time step. WW do not specify what the initial conditions are for the process, so we assume $r_0 = r_{\text{mean}}$, and we then calculate r_i for each $i \in [1, T]$.

The ZI traders use the estimated terminal fundamental value of the security in their trading strategy. We model the fundamental value logic as a function of the security, which the ZI agents call. The estimate is based on the fundamental value r_t at the time t the trader looks it up. The estimated terminal value r_t^{est} is then the expected fundamental value at time T , accounting for the mean-reversion process. WW do not specify how r_t^{est} is made to be an integer, nor whether each component to the price should be converted to int or just the final price. We assume the estimated terminal value should be rounded after the calculation of r_t^{est} . In other words,

$$r_t^{\text{est}} = \text{round} \left(\left(1 - (1 - \kappa)^{T-t} \right) r_{\text{mean}} + (1 - \kappa)^{T-t} r_t \right). \quad (3.2)$$

3.3.6 LATENCY ARBITRAGEUR (LA) TRADER

The latency arbitrageur (LA) agent enacts a cross-market arbitrage strategy, taking advantage of disparities in price arising from fragmentation of orders across multiple exchanges. At initialization, the LA agent subscribes to the BBO feeds from both exchanges. Over time, it tracks the overall best prices available in the market with zero latency. Each time the LA agent receives a new BBO update from an exchange, it evaluates whether to act. Let BID^i and ASK^i denote the best bid and ask at exchange i at a given point in time. Then let $BID^* = \max\{BID^1, BID^2\}$ and $ASK^* = \min\{ASK^1, ASK^2\}$. The LA agent has a threshold parameter α that determines whether an arbitrage opportunity is worth acting on. If $BID^* > (1 + \alpha)ASK^*$, the LA agent submits an order to buy at the floor of the midpoint price `m_floor`:

$$\text{m_floor} = \left\lfloor \frac{BID^* + ASK^*}{2} \right\rfloor \quad (3.3)$$

at the exchange holding the best ask and an order to sell at the ceiling of the midpoint price `m_ceil`:

$$\text{m_ceil} = \left\lceil \frac{BID^* + ASK^*}{2} \right\rceil \quad (3.4)$$

at the exchange holding the best bid.

3.3.7 ZERO-INTELLIGENCE (ZI) TRADERS

The zero-intelligence (ZI) traders in the model enact a strategy in the family of trader strategies introduced by Gode and Sunder [86]. A ZI trader in the model acts according to its private valuation of the asset at time t based on the common fundamental value r_t and the private marginal benefit for trading with its current net position. WW cite Goettler, Parlour, and Rajan [87] as introducing a similar trading strategy based on marginal private benefits. The trader ‘shades’ its order price by some amount, aiming to achieve surplus value from buying (selling) at a lower (higher) price than its internal valuation of the asset.

Initialization

Each ZI trader agent is assigned a ‘primary’ exchange. We assume the ZI traders are evenly split across the exchanges for a given market configuration. Each ZI trader subscribes to the SIP NBBO and to the BBO feed from its primary exchange. Its holdings, profits, and number of transactions are all initialized to zero. Next, for a given trader ZI_i , we construct Θ_i , the vector of marginal private benefits for changes to the trader’s net position:

$$\Theta_i = \left(\theta_i^{-q_{\max}+1}, \dots, \theta_i^0, \theta_i^{+1}, \dots, \theta_i^{q_{\max}} \right), \quad (3.5)$$

where $q_{\max}$ is the maximum number of shares the trader can be long or short the asset. The vector Θ_i is generated from $2q_{\max}$ independent draws from $N(0, \sigma_{PV}^2)$, sorted such that $\theta^q \geq \theta^{q+1} \forall q$. If ZI_i has position q , buying an additional share yields a marginal private benefit θ_i^{q+1} , while selling a share yields a marginal private benefit θ_i^q .

The initial ZI arrival times $\{a_i\}$ are each drawn independently from an exponential distribution,

$$a_i \sim [Exponential(1/\lambda)]. \quad (3.6)$$

WW do not specify how to make the Poisson process discrete. We choose to take the ceiling in order for individual trader interarrival times to be greater than zero.

Arrival and base trading strategy

Upon arrival at time t , the first thing trader ZI_i does is schedules its next arrival for $t + b_i$, where again,

$$b_i \sim [Exponential(1/\lambda)]. \quad (3.7)$$

Next, the trader cancels any unfilled orders it submitted previously. Then, it randomly decides whether to buy or sell (with equal probability). If the trading decision would result in ZI_i 's net position exceeding $q_{\max}$ shares long or short, it exits and does not submit an order on this turn. Otherwise, the trader calculates its current private valuation $v_i(t)$ of the asset:

$$v_i(t) = r_t^{\text{est}} + \begin{cases} \theta_i^{q_t+1} & \text{if buying one unit} \\ \theta_i^{q_t} & \text{if selling one unit} \end{cases}. \quad (3.8)$$

WW do not specify how the price should be made to be discrete. We choose to round $v_i(t)$ to the nearest integer. The trader then determines its order price p as:

$$p \sim \begin{cases} \max(0, U[v_i(t) - R_{\max}, v_i(t) - R_{\min}]) & \text{if buying} \\ \max(0, U[v_i(t) + R_{\min}, v_i(t) + R_{\max}]) & \text{if selling} \end{cases}, \quad (3.9)$$

where U here denotes an integer uniformly chosen at random within the specified range. The requested surplus for ZI_i is then $s_i(t) = |v_i(t) - p_i(t)|$.

Greedy strategy extension

WW introduce an additional strategy extension, which we refer to as the ‘greedy strategy’⁴ going forward. Below is the entirety of the detail given in the paper about this strategy:

We extend ZI by including a threshold parameter $\eta \in [0, 1]$, whereby if the agent could achieve a fraction η of its requested surplus at the current price quote, it would simply take that quote rather than posting a limit order to the book. Setting $\eta = 1$ is equivalent to the strategy without employing the threshold. [189, p. 75]

Note also that WW separately say,

Background traders have direct access to the quotes on their primary market and the NBBO, but not to those on the alternate market. [189, p. 76]

⁴The ‘greedy’ terminology is taken from the *MarketSim* codebase and not from the original paper.

Thus, in our ‘best guess’ (*BestGuess*) implementation of the model, we implement this strategy as the following: based on both the BBO feed from their primary exchange and the NBBO feed from the SIP, let `max_bid` and `min_ask` be the best market prices observed by ZI_i . The trader then evaluates whether there is a greedy opportunity, updating its price if so:

```

1     requested_surplus = |v_i(t) - p|
2     if side == 'BUY':
3         if requested_surplus * eta <= v_i(t) - min_ask:
4             p = min_ask
5     else: // 'SELL'
6         if requested_surplus * eta <= max_bid - v_i(t):
7             p = max_bid

```

Routing strategy

Trader ZI_i will submit its order to its primary exchange unless the NBBO indicates the order would immediately execute at a better price at the alternate exchange. Specifically, let **BBO** refer to the primary exchange BBO, and assume `order.exchange` is originally set to the ZI_i ’s primary exchange. Then the order routing logic is as follows:

```

1  route_order(order):
2      NBB0_price_better = False
3      will_transact = False
4      if order.type == "BUY":
5          if NBB0.ask.price != Null:
6              if (BBO.ask == Null) or (NBB0.ask.price < BBO.ask):
7                  NBB0_price_better = True
8                  if order.price >= NBB0.ask.price:
9                      will_transact = True
10             alt_ex = NBB0.ask.exchange
11 elif order.type == "SELL":
12     if NBB0.bid.price != Null:
13         if (BBO.bid == Null) or (NBB0.bid.price > BBO.bid):
14             NBB0_price_better = True
15             if order.price <= NBB0.bid.price:
16                 will_transact = True
17         alt_ex = NBB0.bid.exchange
18
19 if NBB0_price_better and will_transact:
20     order.exchange = alt_ex
21
22 order.exchange.submit_order(order)

```

3.3.8 MODEL EXPERIMENTS

Environment	N_{ZI}	λ	κ	T
1	24	0.05	0.05	15000
2	238	0.005	0.02	10000
3	58	0.005	0.02	5000

Table 3.4: Parameter settings for the three market environments from WW.

Parameter	Default	Description
r_{mean}	100,000	Mean fundamental value of security.
σ_{shock}^2	5,000,000	Variance of the random shock to the security's fundamental value.
σ_{PV}^2	5,000,000	Variance of ZI private valuations; used to draw private marginal utility of the agent's net position.
α	0.001	Threshold from which the LA agent determines whether an opportunity is worth pursuing.
q_{max}	10	Maximum position (long or short) for ZI traders

Table 3.5: Default parameters used across model configurations. These were specified by WW except for α , which is based on the 2013 paper [190] and MarketSim.

WW conduct experiments across three environments. The parameters varied across these environments are shown in Table 3.4. The parameter N_{ZI} is the number of ZI traders, λ is the arrival rate of ZI traders, κ is the strength of mean reversion of the security's fundamental value, and T is the simulation length. The fixed parameters across environments are shown in Table 3.5. Within an environment, there are three possible configurations of the market: single exchange (*CDA*); two exchanges with no LA agent (*2M no LA*); and two exchanges with LA agent (*2M LA*). In the two-market configurations, latency δ is also varied, over a different set of values for each environment. Specifically:

Environment 1: $\delta \in \{0, 100, 200, 300, 400, 600, 700, 900\}$

Environment 2: $\delta \in \{0, 50, 100\}$

Environment 3: $\delta \in \{0, 25, 50, 75, 100\}$

An experiment in the model is therefore defined by the environment, market configuration, and latency setting.

ID	$R_{\min}$	$R_{\max}$	η
ZI_1	0	125	1
ZI_2	0	250	1
ZI_3	0	500	1
ZI_4	250	500	1
ZI_5	0	1000	1
ZI_6	500	1000	0.4
ZI_7	500	1000	1
ZI_8	0	1500	0.6
ZI_9	1000	2000	0.4
ZI_{10}	0	2500	0.4
ZI_{11}	0	2500	1

Table 3.6: *ZI strategy combinations, taken from WW [189, p. 80].*

The allowed ZI strategies considered by WW are given in Table 3.6. In a run of the model, each ZI agent has a fixed strategy from this set of options. To narrow down the space of possible ZI strategy assignments, WW use an approach called ‘empirical game-theoretic analysis’ (EGTA) [39, 192]. They assume a given experimental setting is a game in which the ZI agents aim to maximize their expected payoff (surplus) based on the strategies of other players. A ‘strategy profile’ specifies a probability vector over the ZI strategies. WW approximate the full game with fewer players to find the Nash equilibria strategy profiles for the game. Then, they simulate the full game by independently sampling a ‘mixture’ of ZI strategies from the equilibrium strategy profile that maximizes total ZI surplus. A mixture is a draw of strategy assignments for all N_{ZI} agents from the equilibrium profile; for each mixture, 100 independent simulation runs are conducted. For a given experiment, they sample 500 mixtures, thus running the model 50,000 times for each experiment.

WW note that limiting their main analysis to these selected profiles allows them to “focus on trader behavior in equilibrium, when all market participants are best

responding to each others’ strategies in order to optimize their own gains from trade” [189, p. 77]. We focus on these same ‘equilibrium’ scenarios in order to assess the fidelity of our replication to the results presented by WW. We therefore do not repeat WW’s EGTA here and instead take their selected strategy profiles as specified. We summarize these strategy profiles for each experiment in our supplemental material in Table B.10. We sample mixtures from these profiles and run simulations similar to the main results presented by WW. We sample 5,000 mixtures and simulate 100 runs per mixture for each experiment, as discussed further in Section 3.4.1.

To summarize, the model experiments are run as follows, with

$\{M = 500, R = 100\}$ for WW and $\{M = 5000, R = 100\}$ for our replication attempts:

1	For each experiment (environment+configuration+latency):
2	Sample M strategy mixtures from equilibrium strategy profile
3	For each mixture m:
4	Run R independent simulations

3.3.9 OUTPUT METRICS

WW calculate the mean value for each metric taken across 50,000 simulations (500 mixtures, 100 runs per mixture) for each experiment. The metrics are calculated and saved at the end of each simulation. The means are calculated on the saved values across simulations (i.e., averaged over all 50,000 runs). WW report the quantitative value for the mean ZI trader surplus and LA surplus (if any) for each experiment. All other metrics are reported visually.

Surplus

At the end of the simulation, a ZI agent ZI_i will have a net terminal position $q_i^T \in [-q_{\max}, q_{\max}]$. The terminal surplus S_i for ZI_i is the sum of the public and private value of its net position. Let r_T be the fundamental value of the asset at time T , and let $\Theta_i = \{\theta_j\}$ be the marginal private benefits vector for ZI_i constructed in Eq. 3.5. Also, let c_i be the net cash flow of ZI_i from its trades over the course of the simulation. Terminal surplus S_i is then calculated at the end of the simulation as:

$$S_i = c_i^T + q_i^T r_T + \begin{cases} \sum_{j=1}^{q_i^T} \theta_i^j & \text{if } q_i^T > 0 \\ 0 & \text{if } q_i^T = 0 \\ \sum_{j=q_i^T+1}^0 -\theta_i^j & \text{if } q_i^T < 0 \end{cases} \quad (3.10)$$

The surplus for the LA agent is simply its net cash flow from trading. This is because the LA agent never has a nonzero net position after it is done executing its strategy; it buys and sells one unit of the asset each time it acts.

Spread

The spread is the amount by which the best ask price (*ASK*) exceeds the best bid price (*BID*); in other words, $spread = ASK - BID$. WW take the median spread over the course of the simulation, then take the mean over all simulations. The median NBBO spread is the median amount the NBBO ask exceeds the NBBO bid over the simulation. The BBO spread is the median spread from the exchange BBO. When there are two exchanges, this is the mean of the median spreads from the individual exchange BBOs.

WW do not address how to handle a locked or crossed NBBO (i.e., where $ASK \leq BID$) when calculating spread, nor what to do when there are no quotes on one side of the book. WW also do not clarify whether the median is calculated on the spread for each quote update or if the spread should be calculated once per time step. We assume it should be calculated for each update (i.e., tick time, rather than clock time). The NBBO median spread is thus taken on the spreads calculated for each NBBO update where the BID and ASK are both non-empty and $BID \leq ASK$. Similarly, we calculate the exchange-level BBO median spread as the median of all spreads on BBO updates where the BID and ASK are both non-empty ($BID < ASK$ at the exchange-level by definition of the CDA), and the medians are averaged across all exchanges in the market.

Execution time

WW describe execution time as ‘the interval between order submission and transaction for orders that eventually trade’ [189, p. 79]. We assume the execution time should be calculated for both legs of the trade, therefore meaning at least one leg of each trade will have an execution time of zero. The mean execution time is then the average of the execution times minus submission times for all orders that resulted in trades.

Transactions

Finally, WW present the number of transactions from the model but do not elaborate on how they calculate this. Based on Fig. 11 in the original study [189, p. 90] and the behavior of the model, we infer the transactions reported by WW are the total

number of *orders* that resulted in trades throughout the simulation, separated by the type of trader that submitted the order. A trade between two ZI traders thus results in two ZI transactions, while a trade between a ZI trader and the LA agent is logged as one ZI transaction and one LA transaction.

3.4 QUANTITATIVE ALIGNMENT

A critical piece of the replication process is having a clear and robust replication standard—the criteria used to determine whether replication has been achieved to the desired level [13, 193]. This requires a reliable measure to assess alignment between two sets of results. As discussed in Section 3.2, Axtell et al. [13] provide a framework to evaluate the extent two model implementations produce equivalent behavior. The amount and type of information provided by the original study determines what tests of equivalence can be employed by the replicator [151].

We do not have the random seeds used by WW for their simulation runs and thus inherently rule out achieving numerical identity of results. The next level of equivalence in the Axtell et al. [13] framework is distributional equivalence. This requires distributional information about the reported measurements. In our case, the only quantitative information reported by WW are the mean ZI surplus values and the mean LA surplus (if any) from the 50,000 simulations (500 mixtures, 100 runs per mixture) for each experiment. Other results in the paper are reported graphically, allowing us to test relational equivalence but not quantitative alignment.

The mean surplus values provide quantitative information we can test alignment against beyond just relational equivalence, however. For example, two sets of results

could be relationally equivalent while being an order of magnitude different quantitatively from one another. The amount of information provided by WW falls somewhere between the level of detail needed for relational and distributional equivalence. In theory, two models could produce sample means indistinguishable from one another while having statistically distinguishable distributional details (e.g., variance) if that information was known. We will refer to such checks simply as quantitative alignment going forward, with the understanding that what we are able to test here falls short of distributional equivalence. We develop our quantitative alignment methodology in the next section before applying it to test different implementations of the model against the canonical results.

3.4.1 METHODOLOGY

In order to make as complete a use of the source information provided as possible, we seek to determine a meaningful test of quantitative alignment against the mean surplus values before moving on to assessing relational equivalence. We have the sample mean values from WW, while we have the full set of results from running our implementations of the model. A naive approach to test quantitative alignment in this circumstance is to repeat their analysis and run our code for 500 agent mixtures (100 runs per mixture) and then test our distribution of values against the reported mean from WW. One possibility is to apply a one-sample t-test to test the hypothesis that our mean surplus value differs from the WW mean. This is a strong assumption: it treats the sample mean reported by WW as if it is approximately the population mean for surplus values generated by the model for the specified experimental setting. Our criteria could then be to reject quantitative alignment for a given experimental

setting if the t-test p-value is below a selected threshold ρ .

We validate our alignment methodology by testing our implementation of the model against itself before comparing it to WW. As we do not have the random seeds used by WW, our measure of alignment should ideally not fail due to simply using a different random seed if all other logic is the same. In order to have confidence in our measure of quantitative alignment, our expectation is the following: if we apply the test to two sets of results produced by the exact same code but with different random seeds, we should reject quantitative alignment approximately $100\rho\%$ of the time. To test this expectation, we ran our *BestGuess* model for 5,000 mixtures, 100 runs per mixture (500,000 runs in total) for each experiment (environment+configuration+latency).

env	model	latency	t-test rej. %		bootstrap rej. %	
			$\rho = 0.05$	$\rho = 0.01$	95% CI	99% CI
1	CDA	0	24.70	13.00	5.60	1.80
1	2M no LA	0	19.80	7.10	6.40	2.50
1	2M no LA	100	36.10	22.60	6.70	2.10
1	2M no LA	200	38.50	25.00	7.70	2.10
1	2M no LA	300	27.10	13.40	7.80	2.40
1	2M no LA	400	39.30	27.70	5.90	1.70
1	2M no LA	600	24.00	12.90	6.70	1.70
1	2M no LA	700	22.20	12.20	6.70	1.60
1	2M no LA	900	19.20	8.10	7.10	1.80
1	2M LA	100	38.00	23.20	6.30	1.30
1	2M LA	200	41.40	26.10	7.80	1.80
1	2M LA	300	35.90	22.50	5.40	1.10
1	2M LA	400	31.40	18.30	5.90	1.70
1	2M LA	600	17.80	8.40	5.70	0.90
1	2M LA	700	48.80	35.90	7.00	1.60
1	2M LA	900	17.70	6.90	6.90	1.80
2	CDA	0	17.80	7.70	5.40	1.60
2	2M no LA	0	18.00	9.30	6.80	1.70
2	2M no LA	50	17.50	7.60	5.80	1.40
2	2M no LA	100	19.00	7.80	6.70	1.80
2	2M LA	50	18.40	8.00	7.50	1.30
2	2M LA	100	30.00	15.40	5.80	1.80
3	CDA	0	31.10	18.00	5.50	1.20
3	2M no LA	0	18.30	8.50	6.50	1.30
3	2M no LA	25	17.00	7.60	6.80	1.60
3	2M no LA	50	18.00	6.30	6.60	0.90
3	2M no LA	75	20.80	9.00	6.60	1.70
3	2M no LA	100	21.10	8.10	6.10	1.80
3	2M LA	25	32.40	18.90	4.70	1.30
3	2M LA	50	21.00	10.40	6.20	1.60
3	2M LA	75	29.70	15.70	7.70	1.70
3	2M LA	100	33.20	18.70	5.30	1.70

Table 3.7: False rejection rates from testing whether our BestGuess model is quantitatively aligned to itself based on the ZI surplus values, using two different quantitative alignment methodologies. We ran our model for 5,000 mixtures, 100 runs per mixture, for each experiment (environment+configuration+latency). Rejection rates are from 1,000 self-tests. We reject quantitative alignment under the one-sample t-test methodology if the t-test p-value is below the selected threshold ρ . Under the bootstrap confidence interval (CI) methodology, we reject quantitative alignment if the target mean falls outside the selected CI (95% or 99%).

For a test, we randomly select results from 500 mixtures (100 runs per mixture) as our target results x_{target} . Let μ_{target} be the mean ZI surplus value from x_{target} . We then randomly select, with replacement, 500 mixtures from the remaining 4,500 mixtures and let that be our comparison sample x^* . We then check whether we would reject quantitative alignment between the ZI surplus values from x^* and the target mean μ_{target} using the t-test criteria given above for some threshold. We repeat this reliability test 1,000 times, each time selecting a new target mean and comparison sample.

The results are shown in Table 3.7 for thresholds of $\rho = 0.05$ and $\rho = 0.01$. Note that, for each experiment, the ZI surplus target mean and sample results in this test were produced by the same code with the same parameters except for the simulation random seeds. We nevertheless see that under this one-sample t-test methodology we reject *BestGuess* being quantitatively aligned to itself 17.0–48.8% of the time for $\rho = 0.05$ and 6.3–35.9% of the time for $\rho = 0.01$. This rejection rate is quite variable and much higher than expected for the specified significance levels. Based on these results, we may falsely reject two sets of results from the same code being quantitatively aligned one-quarter to almost one-half of the time. We therefore determine we need a better measure of quantitative alignment that accounts for the variability of the target results before we can reliably test against the original model.

We next consider an alternative quantitative alignment criteria, this time using a bootstrap approach. Such an approach was suggested by Axtell et al. when there is “no solid reason to assume a convenient (e.g., Gaussian) form of the underlying distributions” [13, p. 140]. We can utilize the full set of results from running our *BestGuess* implementation for 5,000 agent mixtures, 100 runs per mixture. To con-

struct a bootstrap sample, we randomly select, with replacement, the results from 500 of these mixtures. Constructing 1,000 bootstrap samples, we can calculate a percentile interval over the distribution of 500-mixture sample means. We can then check whether the target mean value falls within this bootstrap confidence interval. If the target mean falls outside this interval, we reject the hypothesis that the surplus values produced by the two versions of the model are quantitatively aligned.

We again sanity check this approach for our *BestGuess* implementation against itself for the mean ZI surplus values. For a single test, we randomly select 500 mixtures (100 runs per mixture) to calculate our target mean μ_{target} . We then apply the updated bootstrap methodology, testing whether the target mean falls within the 95% confidence interval (CI) calculated from 1,000 bootstrap samples from the remaining 4,500 mixtures. We repeat this test 1,000 times, randomly selecting a new target and sample population each time. The results of this test are again shown in Table 3.7. Under the bootstrap methodology the false rejection rate (4.7–7.8%) for the 95%-CI threshold is much lower and less variable than under the one-sample t-test methodology, albeit still slightly higher on average than the expected 5%. This gives us greater confidence in the bootstrap confidence interval methodology than the t-test, as we want to minimize the risk we falsely reject quantitative alignment while still meaningfully comparing our simulated results to the canonical values.

3.4.2 *BESTGUESS* VERSUS WW

Having validated our bootstrap quantitative alignment methodology, we now apply it to test our results against the canonical WW values. As mentioned in the previous section, we ran our *BestGuess* replication model for 5,000 mixtures, 100 runs per mixture,

for a total of 500,000 runs per experiment (environment+configuration+latency). We apply our bootstrap quantitative alignment test for these results against the mean ZI and LA surplus values reported by WW. Using the full set of results, we construct 1,000 bootstrap samples of 500 mixtures, 100 runs per mixture, each. To construct a bootstrap sample of our results, we randomly select, with replacement, a sample of 500 mixtures from the 5,000 total mixtures. We calculate the mean surplus value (ZI or LA depending on which we are testing) for the selected bootstrap sample. Repeating this 1,000 times, we get a range of means produced by our implementation of the model for a given experiment.

Table 3.8 reports the 500-mixture mean ZI surplus values reported by WW, the mean of our bootstrap sample means, the bootstrap standard error (SE), and the 95% and 99% confidence intervals for the difference between our bootstrap sample means and the reported WW mean. We highlight in bold the experiments where the bootstrap confidence intervals contain the original mean, indicating where quantitative alignment is not rejected. We reject quantitative alignment for each of the fragmented market configurations but not for the consolidated *CDA* experiments, for which the target means fall within the 95% confidence intervals. The canonical means are also further away from our range of results when latency is nonzero in the fragmented market configurations.

Table 3.9 similarly reports the 500-mixture mean LA surplus values reported by WW, the mean and SE of our bootstrap sample means, and the 95% and 99% confidence intervals for the difference between our bootstrap sample means and the reported WW mean. Note that these values are limited to the experiments where there is an LA agent. We see further difference between our implementation and the orig-

inal results here. In general, our mean LA surplus values are much lower than the values reported by WW. Note that our ZI surplus values exceed those reported by WW in Table 3.8 for the experiments with LA, and by a larger margin than in the experiments with nonzero latency without LA. Thus, we reject quantitative alignment for all but the simplest configuration of the model (*CDA*), and we appear to get further away with increased environmental complexity.

3.4.3 USING AN EXISTING CODEBASE: *MARKETSIM*

WW note that the market simulation system used in the original study “has been extended by other members of the Strategic Reasoning Group at the University of Michigan and employed in several other studies” [189, p. 78]. The Strategic Reasoning Group made the codebase *MarketSim*⁵ available in 2016, and it was cited in a 2017 study by Wah, Wright, and Wellman (WWW) [191] on the effects of market making in a CDA model. Wah’s dissertation [188] describes common modeling assumptions for WW and WWW and does not appear to differentiate these models aside from the following: the first examines latency arbitrage in a market with multiple exchanges, while WWW looks at market making scenarios in a market with a single exchange. The ZI agent strategies in particular are described with common assumptions and logic between the two models. *MarketSim* also features a SIP and is configurable to run with multiple exchanges and SIP latency. We therefore downloaded the *MarketSim* codebase to try to reproduce results from WW as well as WWW.

MarketSim is mainly coded in Java, with Python wrapper scripts to create the

⁵<https://github.com/egtaonline/market-sim/tree/marketsim1>

env	model	latency	WW ZI	<i>BestGuess</i> Bootstrap ZI Surplus			
			mean	mean	SE	95% CI diff.	99% CI diff.
1	CDA	0	10383	10424.98	25.16	(-8.56, 91.70)	(-19.40, 105.87)
1	2M no LA	0	11807	11646.95	20.46	(-201.47, -119.01)	(-213.62, -108.64)
1	2M no LA	100	10373	12352.73	30.11	(1917.68, 2039.47)	(1898.85, 2055.08)
1	2M no LA	200	10621	12432.74	28.11	(1759.90, 1865.58)	(1745.67, 1883.28)
1	2M no LA	300	11244	13334.28	22.59	(2048.73, 2133.00)	(2034.82, 2144.56)
1	2M no LA	400	10438	12011.45	31.21	(1514.59, 1631.01)	(1494.74, 1646.98)
1	2M no LA	600	11128	13219.34	23.00	(2044.96, 2137.10)	(2032.95, 2147.65)
1	2M no LA	700	11302	13416.63	21.97	(2073.08, 2158.08)	(2067.02, 2174.30)
1	2M no LA	900	12358	14370.30	19.96	(1973.33, 2052.03)	(1960.74, 2066.15)
1	2M LA	100	5919	11087.45	31.05	(5108.51, 5227.72)	(5092.44, 5246.67)
1	2M LA	200	6358	10914.22	28.98	(4499.37, 4613.63)	(4483.80, 4636.16)
1	2M LA	300	6398	11007.58	28.25	(4552.43, 4663.06)	(4539.36, 4678.94)
1	2M LA	400	6130	12006.76	25.31	(5826.29, 5923.12)	(5810.17, 5941.75)
1	2M LA	600	7459	13077.26	19.19	(5581.28, 5655.00)	(5573.17, 5671.59)
1	2M LA	700	5256	9894.58	34.86	(4568.50, 4706.73)	(4550.74, 4722.00)
1	2M LA	900	6819	13113.03	20.10	(6256.27, 6334.16)	(6241.98, 6345.99)
2	CDA	0	136140	136131.59	65.51	(-132.27, 118.40)	(-182.89, 152.51)
2	2M no LA	0	134339	134497.56	68.15	(30.47, 288.08)	(1.55, 330.69)
2	2M no LA	50	135789	152272.08	60.61	(16359.26, 16598.50)	(16321.55, 16624.47)
2	2M no LA	100	136542	150308.53	63.92	(13628.88, 13885.56)	(13593.96, 13918.30)
2	2M LA	50	133177	151116.85	63.00	(17815.33, 18057.21)	(17785.51, 18092.99)
2	2M LA	100	124012	139444.74	79.90	(15269.38, 15587.25)	(15223.51, 15648.18)
3	CDA	0	27482	27472.00	40.48	(-92.02, 66.62)	(-107.86, 90.36)
3	2M no LA	0	29424	29929.43	31.43	(442.42, 569.11)	(425.19, 588.16)
3	2M no LA	25	29347	32160.12	29.81	(2758.37, 2869.43)	(2738.92, 2893.16)
3	2M no LA	50	29479	32518.84	27.80	(2987.94, 3094.59)	(2972.80, 3115.38)
3	2M no LA	75	29271	32020.52	30.45	(2691.58, 2810.43)	(2678.22, 2821.88)
3	2M no LA	100	29277	31964.32	30.76	(2624.09, 2747.89)	(2609.72, 2765.10)
3	2M LA	25	26612	29915.89	40.24	(3218.13, 3380.89)	(3192.64, 3400.87)
3	2M LA	50	27953	31863.76	31.61	(3849.29, 3971.74)	(3829.06, 3988.38)
3	2M LA	75	26388	30566.72	36.17	(4104.39, 4243.83)	(4085.11, 4273.02)
3	2M LA	100	25070	29576.74	41.90	(4428.30, 4592.02)	(4408.80, 4619.89)

Table 3.8: Test of quantitative alignment of ZI surplus results from our *BestGuess* replication versus WW. Each WW ZI value is the mean ZI surplus reported by WW [189, pp. 83, 85] for 500 mixtures (100 simulation runs each) sampled according to the welfare-maximizing strategy profile. Our results are calculated from 5,000 mixtures (100 runs each) based on the same strategy profile. From these, we construct 1,000 bootstrap samples of 500 mixtures each to produce confidence intervals for the mean difference. We report the mean and standard error of the bootstrap sample means, along with the 95% and 99% confidence intervals for the difference between our bootstrap sample means and the reported WW mean. Confidence intervals that contain zero are shown in **bold**, indicating where quantitative alignment is not rejected.

env	model	latency	WW LA mean	mean	<i>BestGuess</i> SE	Bootstrap LA Surplus 95% CI diff.	99% CI diff.
1	2M LA	100	3487	477.46	4.11	(-3017.45, -3001.26)	(-3020.43, -2999.39)
1	2M LA	200	3164	526.96	5.02	(-2646.79, -2627.22)	(-2650.33, -2624.88)
1	2M LA	300	3224	561.61	5.48	(-2673.32, -2651.84)	(-2675.87, -2648.27)
1	2M LA	400	4018	552.45	4.44	(-3474.20, -3456.73)	(-3476.67, -3453.40)
1	2M LA	600	4349	1194.00	3.72	(-3162.28, -3147.79)	(-3164.54, -3146.08)
1	2M LA	700	2958	395.65	4.66	(-2571.55, -2552.57)	(-2574.25, -2550.72)
1	2M LA	900	4825	1120.33	4.72	(-3713.31, -3695.89)	(-3716.12, -3692.78)
2	2M LA	50	2417	1107.80	2.95	(-1314.93, -1303.54)	(-1317.40, -1302.08)
2	2M LA	100	2888	1662.21	3.87	(-1233.64, -1218.41)	(-1235.39, -1216.72)
3	2M LA	25	538	229.84	1.46	(-310.99, -305.24)	(-311.72, -304.05)
3	2M LA	50	1154	448.30	2.04	(-709.58, -701.78)	(-710.43, -700.34)
3	2M LA	75	1470	556.20	2.38	(-918.63, -909.38)	(-920.13, -907.60)
3	2M LA	100	1763	632.59	2.62	(-1135.45, -1125.67)	(-1137.51, -1123.92)

Table 3.9: Test of quantitative alignment of LA surplus results from our *BestGuess* replication versus WW. Each WW LA value is the mean LA surplus reported by WW [189, pp. 83, 85] for 500 mixtures (100 simulation runs each) sampled according to the welfare-maximizing strategy profile. Our results are calculated from 5,000 mixtures (100 runs each) based on the same strategy profile. From these, we construct 1,000 bootstrap samples of 500 mixtures each to produce confidence intervals for the mean difference. We report the mean and standard error of the bootstrap sample means, along with the 95% and 99% confidence intervals for the difference between our bootstrap sample means and the reported WW mean. No confidence intervals contain zero, and thus quantitative alignment is rejected for each experiment.

JSON model configuration files and bash scripts to kick off the Java processes. The *MarketSim* GitHub repo comes with a build tutorial explaining how to download, build, and run *MarketSim*. The tutorial and repo include example model configurations related to the WWW model experiments, but it does not include all configurations needed to replicate WWW and includes none of the experiment configurations to run the experiments from WW. We forked⁶ *MarketSim* in order to update the Python scripts, add the WW experiment configurations, and create a processing script to format the JSON files output from runs of the model. We made no changes to the Java code containing the *MarketSim* model logic.

To ensure proper implementation, we first ran *MarketSim* with the experimental configurations specified in WWW for the experiments with only ZI agents (i.e., no market maker). This exercise and evidence are detailed more fully in Appendix B.1. Despite some uncertainty about WWW’s experimental design, we confirm quantitative alignment to the original results for seven out of eight experiments using the bootstrap quantitative alignment methodology from Section 3.4.1. We also achieve relational equivalence on two other metrics for almost all experiments. This gives us confidence *MarketSim* is as specified by WWW and that we are correctly interpreting how to run it for a given market configuration and mixture of ZI strategy profiles.

We next ran *MarketSim* for the experiments specified in WW, running the model for 5,000 mixtures, 100 runs per mixture, similar to our independent replication (Section 3.4.2). For each experiment we sample 1,000 bootstrap samples of 500 mixtures, 100 runs per mixture, in order to create 95% and 99% bootstrap confidence intervals. We then compare these confidence intervals to the mean ZI surplus values reported

⁶Our fork is available [here](#).

by WW, which were calculated from a sample of 500 mixtures, 100 runs per mixture. Shown in Table 3.10, we reject quantitative alignment for each experiment. We see large differences between the original ZI surplus results and our results from *MarketSim*; even for the simplest configuration of the model—the consolidated *CDA* settings—the *MarketSim* results are $\sim 10\%$ greater than the results reported by WW.

The quantitative alignment tests for mean LA surplus from *MarketSim* versus WW similarly show differences, shown in Table 3.11. Interestingly, the range of LA surplus values from *MarketSim* are similar to those we saw from our *BestGuess* implementation, with differences versus WW of similar magnitude. The *MarketSim* mean LA surplus values are significantly lower than those reported by WW for all but one experiment (*2M LA* with $\delta = 25$ in Environment 3). This is despite the *MarketSim* ZI surplus values being much larger than those reported by WW. As we were successful at achieving quantitative alignment with *MarketSim* in most experiments from WWW but not WW, these results suggests some difference in model logic between the current codebase and what was used for the WW study.

env	model	latency	WW mean	mean	<i>MarketSim</i> SE	Bootstrap ZI Surplus 95% CI diff.	99% CI diff.
1	CDA	0	10383	12289.73	26.74	(1855.59, 1959.12)	(1846.53, 1973.57)
1	2M no LA	0	11807	14019.45	19.41	(2173.89, 2250.87)	(2163.41, 2263.96)
1	2M no LA	100	10373	11493.66	30.53	(1064.36, 1184.35)	(1040.53, 1201.47)
1	2M no LA	200	10621	11772.13	31.48	(1090.31, 1214.70)	(1070.75, 1236.44)
1	2M no LA	300	11244	12527.27	24.46	(1237.25, 1330.07)	(1223.00, 1339.55)
1	2M no LA	400	10438	11612.09	32.19	(1110.35, 1234.00)	(1091.75, 1259.64)
1	2M no LA	600	11128	12293.56	25.01	(1117.65, 1215.88)	(1105.85, 1227.51)
1	2M no LA	700	11302	12573.54	24.60	(1222.85, 1319.70)	(1213.19, 1336.33)
1	2M no LA	900	12358	13942.99	19.74	(1545.23, 1623.89)	(1534.78, 1637.40)
1	2M LA	100	5919	10320.99	33.90	(4335.02, 4466.33)	(4309.45, 4496.33)
1	2M LA	200	6358	10335.76	31.19	(3913.24, 4038.55)	(3899.93, 4059.57)
1	2M LA	300	6398	10484.96	30.72	(4024.74, 4146.65)	(4010.35, 4161.10)
1	2M LA	400	6130	11163.95	26.74	(4985.30, 5083.69)	(4965.27, 5104.31)
1	2M LA	600	7459	12350.82	20.70	(4852.06, 4932.84)	(4840.08, 4944.95)
1	2M LA	700	5256	9106.39	43.77	(3765.96, 3934.92)	(3733.69, 3961.31)
1	2M LA	900	6819	12370.84	20.19	(5512.02, 5591.30)	(5497.10, 5602.61)
2	CDA	0	136140	152444.71	60.11	(16188.98, 16419.37)	(16149.99, 16452.03)
2	2M no LA	0	134339	153540.75	61.67	(19079.45, 19322.28)	(19030.06, 19360.26)
2	2M no LA	50	135789	153328.55	60.02	(17420.62, 17660.11)	(17377.15, 17679.95)
2	2M no LA	100	136542	152249.01	62.77	(15587.16, 15837.89)	(15549.02, 15884.94)
2	2M LA	50	133177	151447.36	59.22	(18154.99, 18387.69)	(18112.46, 18420.29)
2	2M LA	100	124012	138244.54	90.23	(14047.07, 14410.17)	(13991.34, 14460.46)
3	CDA	0	27482	30428.98	43.63	(2859.82, 3035.40)	(2838.94, 3057.55)
3	2M no LA	0	29424	32620.14	29.97	(3140.55, 3254.04)	(3119.73, 3274.39)
3	2M no LA	25	29347	32249.51	30.11	(2843.98, 2959.56)	(2820.94, 2976.45)
3	2M no LA	50	29479	32432.73	30.05	(2896.54, 3014.19)	(2875.82, 3032.04)
3	2M no LA	75	29271	32122.41	29.37	(2793.07, 2906.29)	(2780.78, 2919.21)
3	2M no LA	100	29277	32162.18	28.95	(2829.27, 2943.19)	(2812.02, 2961.46)
3	2M LA	25	26612	29014.75	46.33	(2310.24, 2489.45)	(2287.27, 2512.28)
3	2M LA	50	27953	31366.24	32.38	(3353.59, 3477.03)	(3332.15, 3502.22)
3	2M LA	75	26388	29840.00	41.25	(3370.17, 3531.57)	(3341.22, 3551.45)
3	2M LA	100	25070	28743.86	45.32	(3587.38, 3757.65)	(3563.13, 3785.74)

Table 3.10: Test of quantitative alignment of ZI surplus from *MarketSim*'s results versus WW. Each WW ZI value is the mean ZI surplus reported by WW [189, pp. 83, 85] for 500 mixtures (100 simulation runs each) sampled according to the welfare-maximizing strategy profile. Our results are calculated from 5,000 mixtures (100 runs each) based on the same strategy profile. From these, we construct 1,000 bootstrap samples of 500 mixtures each to produce confidence intervals for the mean difference. We report the mean and standard error of the 1,000 bootstrap sample means, along with the 95% and 99% confidence intervals for the difference between our bootstrap sample means and the reported WW mean. No confidence intervals contain zero, and thus quantitative alignment is rejected for each experiment.

env	model	latency	WW LA mean	mean	SE	<i>MarketSim</i> Bootstrap LA Surplus 95% CI diff.	99% CI diff.
1	2M LA	100	3487	582.32	6.65	(-2918.07, -2891.66)	(-2922.02, -2887.51)
1	2M LA	200	3164	642.50	7.71	(-2536.68, -2506.74)	(-2542.10, -2502.44)
1	2M LA	300	3224	679.99	7.46	(-2558.63, -2528.98)	(-2564.73, -2525.40)
1	2M LA	400	4018	726.28	6.94	(-3305.61, -3278.20)	(-3309.42, -3274.14)
1	2M LA	600	4349	1488.15	5.86	(-2871.72, -2848.81)	(-2875.02, -2845.20)
1	2M LA	700	2958	437.99	6.62	(-2533.06, -2506.82)	(-2536.59, -2504.53)
1	2M LA	900	4825	1382.05	6.44	(-3455.26, -3430.16)	(-3459.43, -3425.37)
2	2M LA	50	2417	1842.37	3.99	(-582.45, -566.88)	(-585.29, -564.66)
2	2M LA	100	2888	1552.83	3.51	(-1342.13, -1328.32)	(-1343.29, -1326.44)
3	2M LA	25	538	658.55	2.94	(114.82, 126.18)	(113.50, 128.41)
3	2M LA	50	1154	859.93	2.80	(-299.76, -288.58)	(-300.62, -286.63)
3	2M LA	75	1470	841.49	3.10	(-634.26, -622.17)	(-635.90, -620.59)
3	2M LA	100	1763	821.54	3.53	(-948.24, -934.72)	(-950.18, -931.89)

Table 3.11: Test of quantitative alignment of LA surplus from *MarketSim*'s results versus WW. Each WW LA value is the mean LA surplus reported by WW [189, pp. 83, 85] for 500 mixtures (100 simulation runs each) sampled according to the welfare-maximizing strategy profile. Our results are calculated from 5,000 mixtures (100 runs each) based on the same strategy profile. From these, we construct 1,000 bootstrap samples of 500 mixtures each to produce confidence intervals for the mean difference. We report the mean and standard error of the bootstrap sample means, along with the 95% and 99% confidence intervals for the difference between our bootstrap sample means and the reported WW mean. No confidence intervals contain zero, and thus quantitative alignment is rejected for each experiment.

3.4.4 DIFFERENCES BETWEEN *MARKETSIM* AND OUR *BESTGUESS* IMPLEMENTATION

Given the overlap in timing and authorship for *MarketSim* and the WW model, we felt it may hold insights for the replication despite also having apparent differences. Not knowing what differs between the two versions of the model makes it difficult to determine what design choices from *MarketSim* are (1) necessary to recreate the original results, (2) responsible for the differences in results, or (3) immaterial. We identified two key differences between the *MarketSim* logic and our *BestGuess* imple-

mentation. The first is a difference in the information used by the ZI traders for their ‘greedy strategy’. The second is an apparent bug in the order routing logic.

Greedy strategy

The ZI agent strategy used in the WW and WWW model is referred to as ‘ZIRP’ (zero-intelligence with re-submission plus) in *MarketSim*. Its base strategy⁷ appears to be implemented similar to our *BestGuess* logic detailed in Section 3.3.7–3.3.7. A difference is found when we examine the greedy strategy logic, however. Below is the greedy logic from *MarketSim*. Let **BBO** again refer to the BBO for the trader ZI_i ’s primary exchange, $v_i(t)$ denote private valuation of the asset, p the order price from the base strategy, and η the trader’s greedy threshold parameter:

```

1    max_bid = BBO.BID
2    min_ask = BBO.ASK
3    requested_surplus = |v_i(t) - p|
4    if side == 'BUY':
5        if requested_surplus * eta <= v_i(t) - min_ask:
6            p = v_i(t)
7    else: // 'SELL'
8        if requested_surplus * eta <= max_bid - v_i(t):
9            p = v_i(t)

```

There are two key differences to note between the above logic and our *BestGuess* interpretation (Section 3.3.7). First, in *MarketSim* the trader only uses the primary exchange BBO to determine if a greedy opportunity exists; it does not look at the NBBO. Second, if a greedy opportunity exists, the trader sets its price to be its *private valuation* $v_i(t)$, rather than equal to the BBO *ASK* or *BID*. In a single-exchange market (inherently with no latency), this logic will be equivalent to our interpretation, as the *BBO* and *NBBO* are equivalent, and the greedy buy (sell) order will execute

⁷The core ZIRP strategy, including its greedy strategy logic, can be found in the *MarketSim* code.

at the *ASK* (*BID*) price. However, if there are two exchanges, greedy opportunities at the alternative exchange will not be acted on in the *MarketSim* logic. If there are two exchanges and latency is nonzero, the order may be re-routed to the alternative exchange based on stale information, potentially resulting in the worst case scenario where the order executes (or becomes a limit order) with price $v_i(t)$, resulting in zero achieved surplus for the trader. Setting the order price to $v_i(t)$ is overly aggressive if the goal is to achieve at least fraction η of its requested surplus $s_i(t)$ (i.e., $\eta s_i(t)$).

WW say the ZI agents have access to quotes on their primary market and the NBBO. It would seem strange for the ZI agents to then only use the primary market quotes to inform their greedy order strategy, particularly if a greedy order can still be rerouted to the alternative exchange if the NBBO specifies a better price there. WW also say, “Setting $\eta = 1$ is equivalent to the strategy without employing the threshold” [189, p. 75]. However, in the *MarketSim* implementation, a trader with $\eta = 1$ could still end up submitting a greedy order at price $v_i(t)$ that then gets misrouted to the other exchange due to a stale NBBO. Thus, setting $\eta = 1$ is not equivalent to not employing the greedy strategy in this implementation.

Order routing

The other main difference we found between *MarketSim* and our interpretation of the model is the logic used to route ZI orders. *MarketSim* has the ZI agent’s primary exchange reroute the order based on the NBBO, rather than having the trader make the routing decision. This makes some sense, as exchanges in the NMS are required to reroute orders based on best-execution rules [179]. At the same time, WW’s description of the model implies the traders do the routing. Also, if the trader does

not use the SIP NBBO in its pricing strategy, and it does not use it to route its own order, why does the trader need access to the NBBO? However, there is no latency between a trader submitting an order and the exchange receiving it in the model, and as such the exchange versus the trader doing the routing should not have any effect.

Within the exchange order routing logic in *MarketSim*, however, we find a possible bug. Below (in its original Java) is the `submitNMSOrder` function called by the ZIRP traders in *MarketSim* when they submit an order to their primary exchange:

```

1  public void submitNMSOrder(Agent agent, OrderType type, Price price,
2      int quantity, Timestamp currentTime, Timestamp duration) {
3      checkNotNull(type, "Order type");
4      checkArgument(quantity > 0, "Quantity must be positive");
5
6      BestBidAsk nbbo = sip.getNBBO();
7      Market bestMarket = this;
8
9      if (quantity > 0) { // buy
10         boolean nbboBetter = nbbo.getBestAsk() != null
11             && nbbo.getBestAsk().lessThan(quote.getAskPrice());
12         boolean willTransact = price.greaterThan(nbbo.getBestAsk());
13         if (nbboBetter && willTransact)
14             bestMarket = nbbo.getBestAskMarket();
15     } else { // sell
16         boolean nbboBetter = nbbo.getBestBid() != null
17             && nbbo.getBestBid().greaterThan(quote.getBidPrice());
18         boolean willTransact = price.lessThan(nbbo.getBestBid());
19         if (nbboBetter && willTransact)
20             bestMarket = nbbo.getBestBidMarket();
21     }
22
23     if (!bestMarket.equals(this))
24         log.log(INFO, "Routing %s %s %d @ %s from %s %s to NBBO %s %s",
25             agent, type, quantity, price, this, quote, bestMarket, nbbo);
26
27     bestMarket.submitOrder(agent, type, price, quantity, currentTime,
28         duration);
29 }

```

The exchange checks the NBBO and determines whether the order would execute immediately at a better price at the alternative exchange, rerouting the order if so.

The function contains a check asserting that the order’s **quantity** ‘must be positive’. It then uses the test **quantity > 0** to determine if the order is a buy order as opposed to a sell order. Elsewhere in *MarketSim*, orders have a **type** that equals **OrderType.BUY** or **OrderType.SELL** (see the code referenced in the ZIRP strategy logic above for an example). It appears the **submitNMSOrder** function is using the wrong criteria to determine order type, and in fact it will treat every incoming order as a buy order when determining the routing decision. This means an incoming sell order may be misrouted to the alternative exchange based on the ask prices at that exchange, or it may fail to be rerouted to the alternative exchange when the order may have executed there at a better price than at the primary exchange.

3.4.5 *BESTGUESS+MARKETSIM*

We next took the greedy strategy logic from *MarketSim* highlighted above and integrated it into our *BestGuess* model. We will refer to this implementation of the model as *BestGuess+MS* going forward. All logic in this implementation is as specified in our ODD except for the greedy strategy logic, which now follows the above detail from *MarketSim*. We again ran this model for 5,000 mixtures, 100 runs per mixture, and employ the quantitative alignment methodology outlined in Section 3.4.1.

Our quantitative alignment tests for mean ZI surplus under the *BestGuess+MS* implementation are shown in Table 3.12. We find smaller differences compared to the original results across more experiments in these results than we saw for our *BestGuess* results (Table 3.8). The 95% confidence intervals from *BestGuess+MS* ZI surplus means contain the WW ZI surplus means for the consolidated *CDA* configurations and the fragmented market configurations without latency from Environments

2 and 3. The canonical ZI surplus mean for the fragmented market configuration without latency in Environment 1 is within the 99% confidence intervals for this implementation but not the 95% confidence intervals. We still reject quantitative alignment for all configurations of the model with nonzero latency, however.

Our quantitative alignment results for mean LA surplus under the *BestGuess+MS* implementation are similarly shown in Table 3.13. The differences between the bootstrap sample LA surplus means and the canonical means are much smaller than we saw for either of the previous sets of results from *BestGuess* and *MarketSim*. However, we still reject quantitative alignment for each experiment with LA. Note that the LA achieves much higher surplus on average in the *BestGuess+MS* implementation than in our original *BestGuess* implementation. This shows the latency arbitrage opportunities are more prevalent, larger, or both under the implementation of the ZI greedy strategy taken from *MarketSim* versus our *BestGuess* greedy strategy. This fits with our expectation, as the ZI agents use more of their available information in the *BestGuess* implementation and act more strategically.

A note on the order routing bug

We also considered including the order routing bug from *MarketSim*'s `submitNMSOrder` function, in case this was important to drive the behavior reported by WW. We ran this alternative version of the model (call it *BestGuess+MS+bug*) again for 5,000 mixtures, 100 runs per mixture, and compared against the original results. We overall found that our hybrid version without the bug (*BestGuess+MS*) aligned more closely to the original results than the version with the bug in most experimental settings. We present these results in full in our appendix. For space considerations,

and because it does not appear the bug helps us better replicate the canonical results, we focus on the *BestGuess*, *MarketSim*, and *BestGuess+MS* implementations of the model going forward.

env	model	latency	WW	<i>BestGuess+MS</i> Bootstrap ZI Surplus			
			mean	mean	SE	95% CI diff.	99% CI diff.
1	CDA	0	10383	10425.08	25.05	(-6.59, 91.70)	(-20.88, 107.22)
1	2M no LA	0	11807	11848.60	20.29	(3.85, 79.76)	(-7.32, 91.41)
1	2M no LA	100	10373	10165.29	29.04	(-261.75, -151.72)	(-281.56, -136.20)
1	2M no LA	200	10621	10374.49	29.28	(-304.84, -190.37)	(-319.17, -174.62)
1	2M no LA	300	11244	11066.88	23.35	(-220.67, -129.55)	(-229.91, -113.65)
1	2M no LA	400	10438	10215.79	31.54	(-283.98, -161.31)	(-305.11, -145.09)
1	2M no LA	600	11128	10905.40	23.52	(-269.76, -178.31)	(-286.20, -165.97)
1	2M no LA	700	11302	11132.93	23.40	(-212.51, -124.40)	(-229.52, -110.91)
1	2M no LA	900	12358	12188.92	19.28	(-206.59, -132.95)	(-217.71, -119.99)
1	2M LA	100	5919	4885.89	43.02	(-1117.64, -953.04)	(-1139.36, -920.91)
1	2M LA	200	6358	5311.49	41.96	(-1124.50, -965.41)	(-1161.70, -946.49)
1	2M LA	300	6398	5412.35	40.00	(-1061.06, -906.79)	(-1080.34, -876.78)
1	2M LA	400	6130	5189.03	41.57	(-1022.13, -860.66)	(-1045.77, -831.78)
1	2M LA	600	7459	6510.35	34.76	(-1019.09, -885.71)	(-1037.37, -863.14)
1	2M LA	700	5256	4311.61	48.33	(-1033.21, -844.46)	(-1061.59, -818.67)
1	2M LA	900	6819	6435.05	37.25	(-452.60, -311.27)	(-478.14, -283.28)
2	CDA	0	136140	136130.24	65.87	(-136.49, 118.04)	(-165.45, 156.12)
2	2M no LA	0	134339	134304.59	66.93	(-159.13, 96.80)	(-202.71, 140.73)
2	2M no LA	50	135789	134176.50	67.52	(-1749.85, -1483.62)	(-1784.80, -1445.79)
2	2M no LA	100	136542	134830.23	68.94	(-1847.38, -1579.04)	(-1884.78, -1543.89)
2	2M LA	50	133177	131877.57	66.75	(-1429.96, -1168.87)	(-1454.77, -1117.89)
2	2M LA	100	124012	121634.05	86.93	(-2550.99, -2210.60)	(-2596.45, -2155.82)
3	CDA	0	27482	27473.78	41.34	(-91.20, 70.80)	(-114.83, 99.48)
3	2M no LA	0	29424	29449.31	31.17	(-38.14, 80.64)	(-55.39, 100.15)
3	2M no LA	25	29347	29173.53	30.08	(-232.63, -115.76)	(-245.77, -96.16)
3	2M no LA	50	29479	29256.27	29.72	(-279.13, -164.12)	(-299.10, -148.92)
3	2M no LA	75	29271	29052.93	30.68	(-278.61, -157.04)	(-303.42, -141.71)
3	2M no LA	100	29277	29038.14	29.87	(-295.51, -174.62)	(-315.24, -164.33)
3	2M LA	25	26612	26357.77	40.65	(-335.19, -173.46)	(-365.85, -149.73)
3	2M LA	50	27953	27538.00	36.79	(-485.88, -342.43)	(-508.52, -312.64)
3	2M LA	75	26388	25830.41	41.05	(-640.52, -474.18)	(-659.90, -449.40)
3	2M LA	100	25070	24449.34	46.87	(-715.92, -530.06)	(-738.23, -506.31)

Table 3.12: Test of quantitative alignment of ZI surplus from our *BestGuess+MS* implementation results versus WW. Each WW ZI value is the mean ZI surplus reported by WW [189, pp. 83, 85] for 500 mixtures (100 simulation runs each) sampled according to the welfare-maximizing strategy profile. Our results are calculated from 5,000 mixtures (100 runs each) based on the same strategy profile. From these, we construct 1,000 bootstrap samples of 500 mixtures each to produce confidence intervals for the mean difference. We report the mean and standard error of the 1,000 bootstrap sample means, along with the 95% and 99% confidence intervals for the difference between our bootstrap sample means and the reported WW mean. Confidence intervals that contain zero are shown in **bold**, indicating where quantitative alignment is not rejected.

env	model	latency	WW LA	<i>BestGuess+MS</i> Bootstrap LA Surplus			
			mean	mean	SE	95% CI diff.	99% CI diff.
1	2M LA	100	3487	4279.07	27.57	(738.41, 846.33)	(721.62, 865.13)
1	2M LA	200	3164	3971.93	26.61	(756.94, 855.87)	(738.61, 880.99)
1	2M LA	300	3224	4010.53	27.37	(731.13, 839.26)	(716.75, 853.23)
1	2M LA	400	4018	4771.45	29.85	(694.47, 810.59)	(673.47, 828.50)
1	2M LA	600	4349	5115.70	25.89	(715.43, 816.48)	(701.50, 834.79)
1	2M LA	700	2958	3721.58	26.71	(710.13, 816.54)	(698.10, 830.06)
1	2M LA	900	4825	5113.45	26.97	(235.29, 341.54)	(225.39, 363.58)
2	2M LA	50	2417	2355.86	8.93	(-78.15, -43.91)	(-82.22, -38.47)
2	2M LA	100	2888	3246.52	23.37	(312.28, 404.25)	(301.52, 419.73)
3	2M LA	25	538	616.55	8.17	(63.29, 94.03)	(58.82, 98.98)
3	2M LA	50	1154	1337.14	14.71	(155.44, 212.36)	(147.12, 225.31)
3	2M LA	75	1470	1787.98	20.78	(277.83, 357.39)	(268.44, 372.37)
3	2M LA	100	1763	2173.46	26.17	(359.47, 462.56)	(340.67, 474.26)

Table 3.13: Test of quantitative alignment of LA surplus results from our *BestGuess+MS* replication versus WW. Each WW LA value is the mean LA surplus reported by WW [189, pp. 83, 85] for 500 mixtures (100 simulation runs each) sampled according to the welfare-maximizing strategy profile. Our results are calculated from 5,000 mixtures (100 runs each) based on the same strategy profile. From these, we construct 1,000 bootstrap samples of 500 mixtures each to produce confidence intervals for the mean difference. We report the mean and standard error of the bootstrap sample means, along with the 95% and 99% confidence intervals for the difference between our bootstrap sample means and the reported WW mean. No confidence intervals contain zero, and thus quantitative alignment is rejected for each experiment.

3.5 QUALITATIVE BEHAVIOR OF THE MODEL

We next look at the qualitative behavior of our implementations of the model (*BestGuess*, *MarketSim*, and *BestGuess+MS*), comparing against the results plotted by WW to assess relational equivalence. WW plot the results across experimental settings for surplus, execution time, spread (BBO and NBBO), and the number of transactions. We present the WW results alongside the results from each of our model implementations for easy comparison. The WW results are reproduced from Wah and Wellman [189] by permission of Sage Publications. Each metric from our simulations

are averaged over all 5,000 mixtures, 100 runs per mixture, for a given experiment, versus 500 mixtures for the WW results.

3.5.1 SURPLUS

We start with the average total surplus values across the different experiments (environment+configuration+latency). As shown in Fig. 3.3, our *BestGuess* results exhibit clear differences from the canonical results. The canonical (WW) results show mixed effects on surplus from fragmentation, and the introduction of the LA agent harms surplus in most settings. Our *BestGuess* results show almost every fragmented market configuration, with or without LA, outperforms the consolidated *CDA* configurations. The only exceptions are the *2M LA* results with $\delta = 900$ in Environment 1 and the *2M no LA* results with $\delta = 0$ in Environment 2. In all other cases, surplus is higher with fragmentation, and nonzero latency results in higher surplus relative to no latency. In most of the nonzero-latency settings, the presence of the LA agent reduces surplus, but surplus is still higher in those settings than the single-CDA market.

The *MarketSim* results achieve relational equivalence in Environments 1 and 3 but not in Environment 2. In the Environment 2 *MarketSim* results, the *2M no LA* with $\delta = 0$ surplus is greater than the *CDA* surplus and decreases as latency is increased. In WW, the *2M no LA* surplus with $\delta = 0$ is lower than surplus in the consolidated *CDA* configuration and increase as latency increases. Our *BestGuess+MS* implementation meanwhile achieves relational equivalence for all three environments; the only relational difference is the *2M no LA* surplus for $\delta = 100$ in Environment 2 is slightly lower than the *CDA* surplus rather than slightly above. Overall, these results add support to our previous evidence that the hybrid *BestGuess+MS* model

implementation is the closest to the original model.

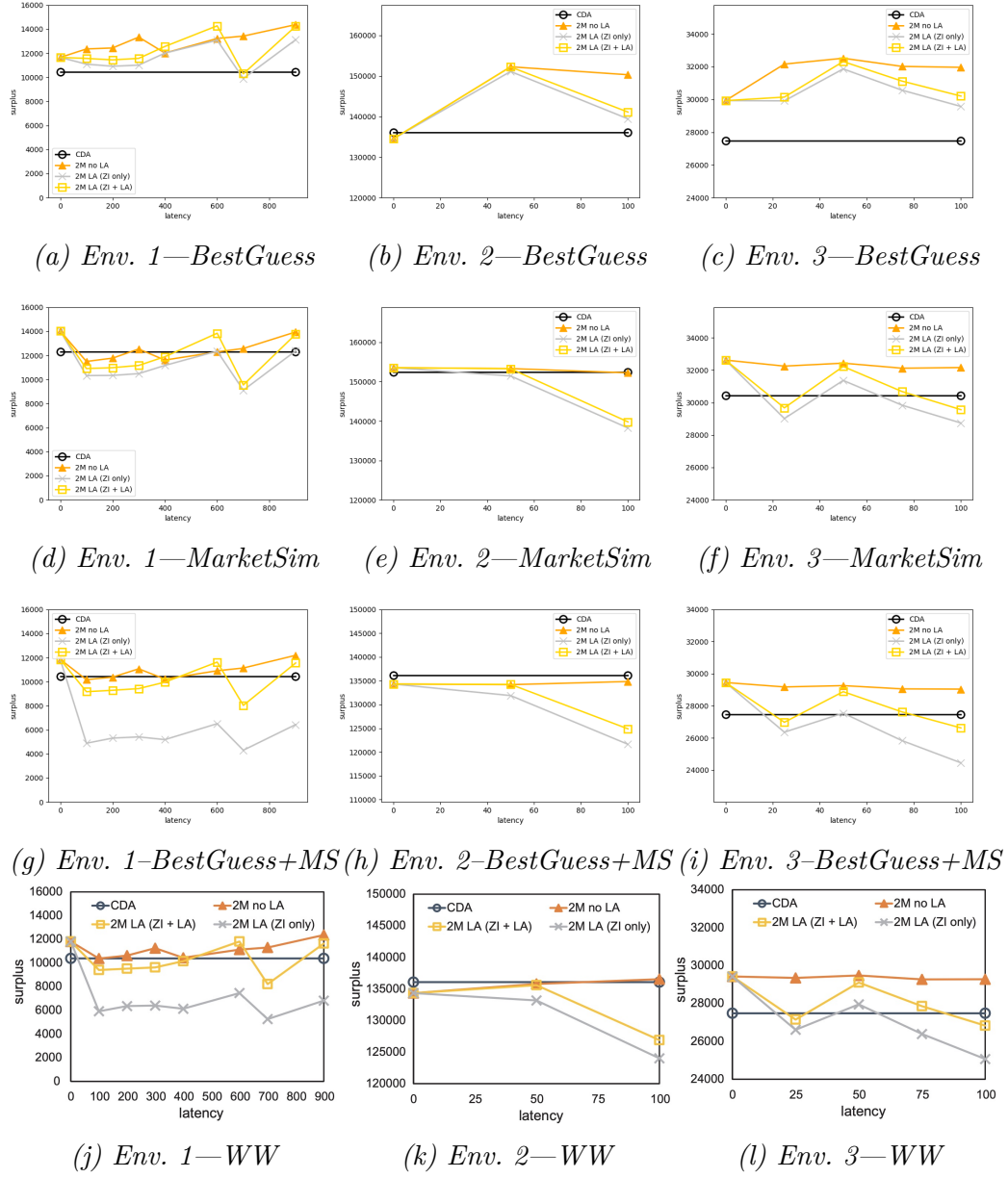
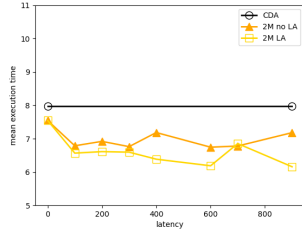


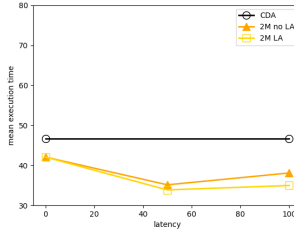
Figure 3.3: Mean surplus results from our implementations of the model and MarketSim versus the figures reported by WW [189, p. 82]. Results from our simulation are averaged over 5,000 mixtures, 100 runs per mixture, for each experiment.

3.5.2 EXECUTION TIME

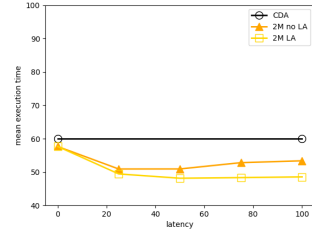
In Fig. 3.4, we show our mean execution time results alongside the figures shared by WW. WW found that execution time in their model was roughly the same or higher with LA than without LA in each experiment except $\delta = 50$ in Environment 3. They also found that fragmentation did not generally improve execution time. In our *BestGuess* implementation results, fragmentation reduces execution time in every configuration of the model, nonzero latency generally improves execution time, and the presence of the LA agent also improves execution time in almost every nonzero latency setting. We therefore again have material differences in qualitative behavior between our *BestGuess* implementation and the original results.



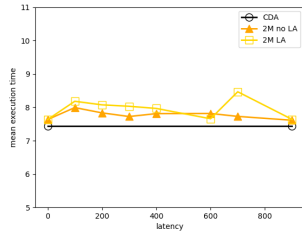
(a) *Env. 1—BestGuess*



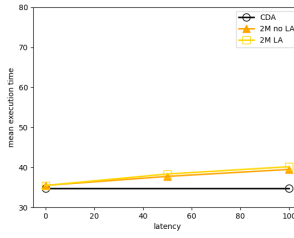
(b) *Env. 2—BestGuess*



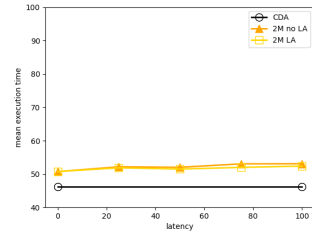
(c) *Env. 3—BestGuess*



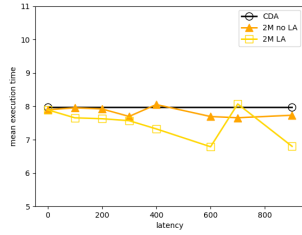
(d) *Env. 1—MarketSim*



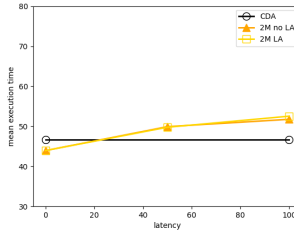
(e) *Env. 2—MarketSim*



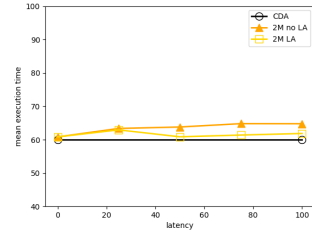
(f) *Env. 3—MarketSim*



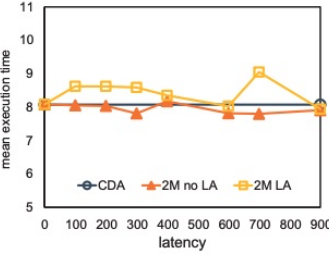
(g) *Env. 1—BestGuess+MS*



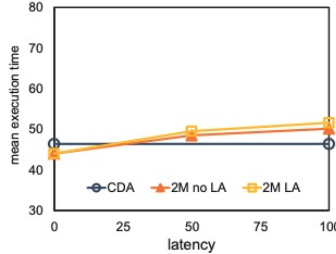
(h) *Env. 2—BestGuess+MS*



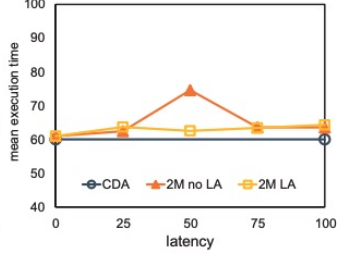
(i) *Env. 3—BestGuess+MS*



(j) *Env. 1—WW*



(k) *Env. 2—WW*



(l) *Env. 3—WW*

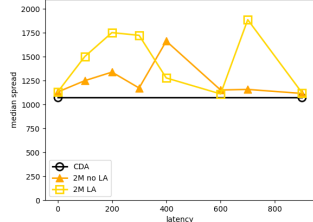
Figure 3.4: Mean execution time results from our implementations of the model and *MarketSim* versus the figures reported by WW [189, p. 86]. Results from our simulation are averaged over 5,000 mixtures, 100 runs per mixture, for each experiment.

Execution time from *MarketSim* has notable differences as well, but in a different direction than our *BestGuess* results. In *MarketSim*, execution time is increased by

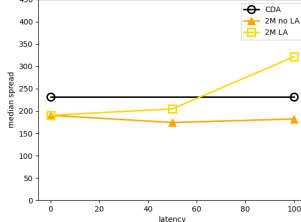
fragmentation relative to the consolidated *CDA* configuration. The presence of the LA agent increases execution time in most cases in Environments 1 and 2, but it has minimal effect in Environment 3. We also see that the *CDA* execution times differ quantitatively from the original results, while the *CDA* results from our other implementations cannot be visually distinguished from WW.

Our *BestGuess+MS* results resemble the WW results in most but not all configurations. In particular, we achieve relational equivalence in Environment 2; we achieve relational equivalence for the *2M no LA* results but not the *2M LA* results in Environment 1; and we achieve relational equivalence for the *2M LA* results but not the *2M no LA* results in Environment 3. Thus, we do not have as consistent of alignment for execution time as we did for surplus between *BestGuess+MS* and WW, but it remains our most faithful replication.

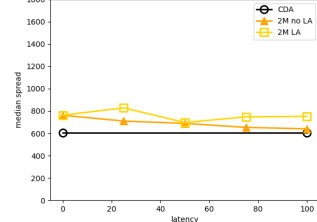
3.5.3 SPREAD



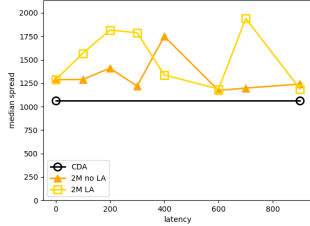
(a) *Env. 1—BestGuess*



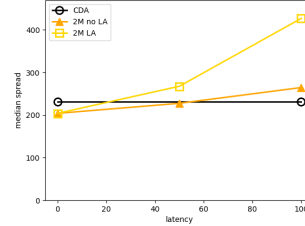
(b) *Env. 2—BestGuess*



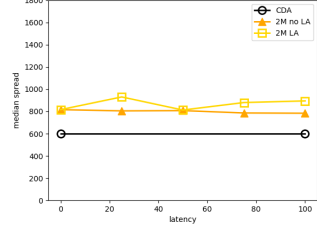
(c) *Env. 3—BestGuess*



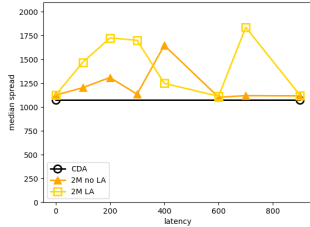
(d) *Env. 1—MarketSim*



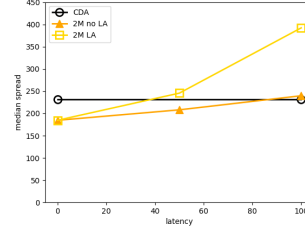
(e) *Env. 2—MarketSim*



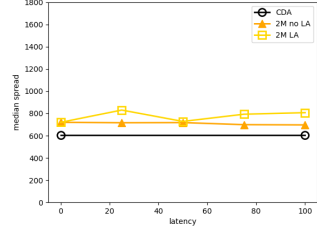
(f) *Env. 3—MarketSim*



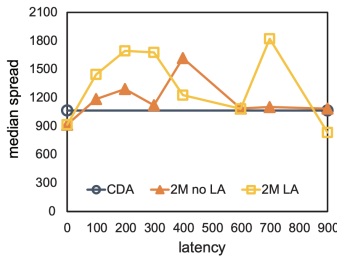
(g) *Env. 1—BestGuess+MS*



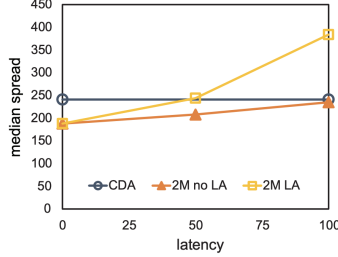
(h) *Env. 2—BestGuess+MS*



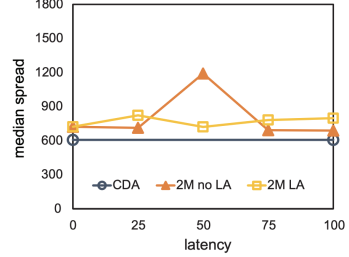
(i) *Env. 3—BestGuess+MS*



(j) *Env. 1—WW*



(k) *Env. 2—WW*

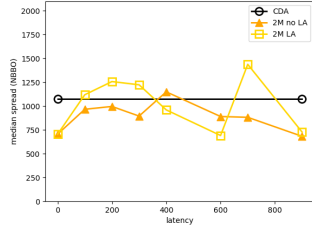


(l) *Env. 3—WW*

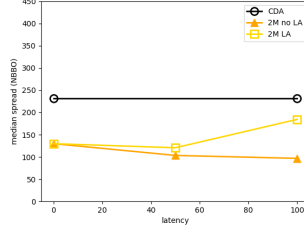
Figure 3.5: Mean median spreads from the exchange BBOs for our implementations of the model and MarketSim versus the figures reported by WW [189, p. 87]. BBO spreads are averaged across the different exchanges in a given simulation run. Results from our simulation are averaged over 5,000 mixtures, 100 runs per mixture, for each experiment.

In Figs. 3.5 and 3.6, respectively, we compare the BBO and NBBO spreads generated by our simulation to the canonical results. In WW, latency arbitrage increases spreads in most settings, while fragmentation and latency in general have mixed effects relative to the consolidated *CDA* market configurations. We are not able to visually distinguish between the *CDA* spreads from WW and our replication results, including for the *MarketSim* results. The presence of the LA agent generally exacerbates BBO spreads in our results, similar to WW. The BBO spreads are otherwise relatively consistent across the three replications, and all differ from the WW results in similar ways (e.g., $\delta = 0$ in Environment 1 and $\delta = 50$ in Environment 3).

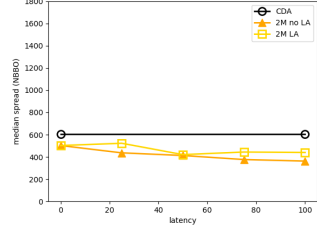
The NBBO spreads for Environment 1 roughly match between each of our replications and WW. Our *BestGuess* results differ somewhat from WW in Environments 2 and 3, with a consistent ranking of the consolidated *CDA* market having the largest NBBO spreads and the *2M no LA* results having the smallest. The relative NBBO spread movements from *MarketSim* and *BestGuess+MS* match WW in Environment 2, while in Environment 3 the with-LA results appear to match WW but the *2M no LA* results do not have the jump in spread at $\delta = 50$ that WW report.



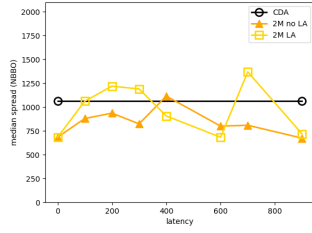
(a) *Env. 1—BestGuess*



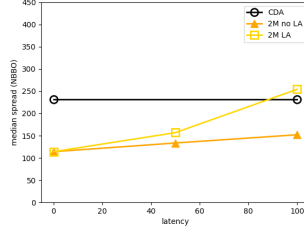
(b) *Env. 2—BestGuess*



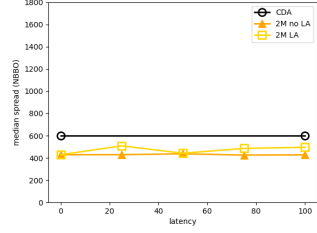
(c) *Env. 3—BestGuess*



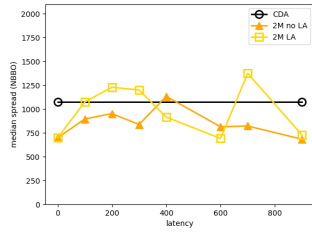
(d) *Env. 1—MarketSim*



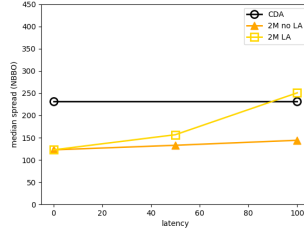
(e) *Env. 2—MarketSim*



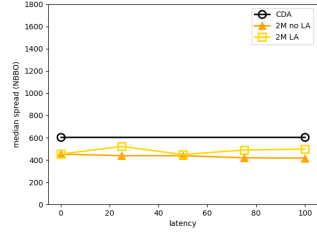
(f) *Env. 3—MarketSim*



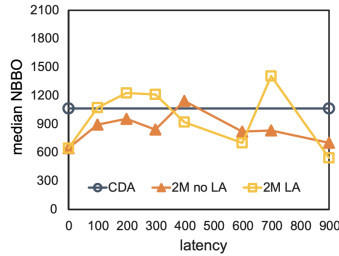
(g) *Env. 1—BestGuess+MS*



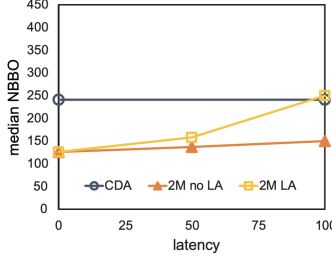
(h) *Env. 2—BestGuess+MS*



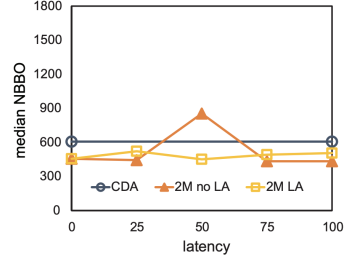
(i) *Env. 3—BestGuess+MS*



(j) *Env. 1—WW*



(k) *Env. 2—WW*



(l) *Env. 3—WW*

Figure 3.6: Mean median SIP NBBO spread for our implementations of the model and *MarketSim* versus the figures reported by WW [189, p. 87]. Results from our simulation are averaged over 5,000 mixtures, 100 runs per mixture, for each experiment.

3.5.4 TRANSACTIONS

Last, we look at the number of transactions generated by our simulation versus WW. As highlighted earlier, it is important to note WW appear to report each trade as two transactions, one for each trader involved in the trade. Thus, a trade between two ZI agents is represented in Fig. 3.7 as two ‘ZI only’ transactions, while a trade between a ZI agent and the LA agent is shown as one ‘ZI only’ transaction and one ‘LA only’ transaction.

Visual appraisal of the relative changes in transaction numbers across experiments and between sets of results is more difficult in the bar chart format, but some differences are apparent between the *BestGuess* results and WW. The zero-latency results appear similar, but, as latency increases, the number of transactions in the *BestGuess* results are lower than those reported by WW. Overall, the fragmented markets without latency have the highest overall levels of trading in our results, while nonzero latency leads to fewer trades in most experiments. The effect of latency arbitrage is mixed, similar to WW’s result. The *MarketSim* and *BestGuess+MS* transaction results track much more closely with the WW results (and with each other). There also appear to be fewer LA transactions in the *BestGuess* results relative to the WW results, fitting with the lower LA surplus values we saw in our quantitative alignment exercise (Table 3.9).

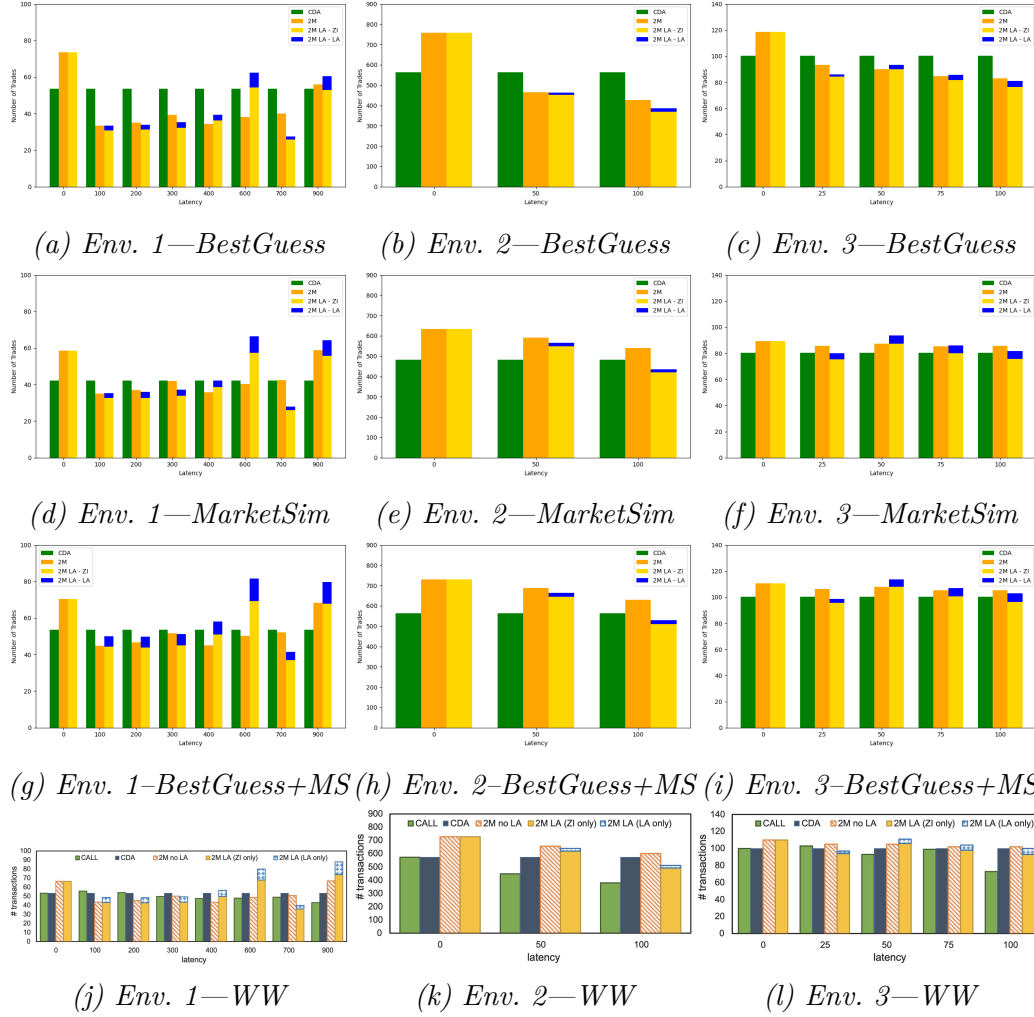


Figure 3.7: Mean number of transactions from our implementations of the model and *MarketSim* versus the figures reported by WW [189, p. 90]. Please note, a single trade results in two transactions for this measurement (one per trader involved in the trade). Results from our simulation are averaged over 5,000 mixtures, 100 runs per mixture, for each experiment.

3.5.5 SUMMARY OF RESULTS

Feature	Effect reported by WW	<i>BestGuess</i> replication
Latency arbitrage	“Generally degrades efficiency, but increased bid shading in low mean reversion environments alleviates inefficiencies caused by LA.”	LA generally degrades efficiency relative to fragmented markets without LA.
Latency arbitrage	“LA exacerbates spreads, and execution times vary by environment based on bid shading in equilibrium”	LA exacerbates spreads but decreases execution times.
Market fragmentation	“Can benefit continuous markets by admitting fewer inefficient trades (due to vagaries of the arrival sequence of orders), but LA defeats this benefit.”	Fragmentation generally improves surplus, with or without LA.
Market fragmentation	“Execution times vary by environment, with consolidation improving liquidity in the fragmented model without LA when traders do not shade more.”	Fragmentation reduces execution times and NBBO spreads, but it increases the average BBO spread in most settings.

Table 3.14: Qualitative summary of model behavior from WW versus our *BestGuess* results. WW summary text is reproduced from Table 5 in Wah and Wellman [189, p. 81].

In the original paper, WW provide a summary of qualitative takeaways from their experimental results. We reproduce their takeaways in Table 3.14 and compare against our *BestGuess* replication. WW report that latency arbitrage reduces total surplus and has a mixed effect on liquidity, while eliminating fragmentation can improve surplus and liquidity metrics. The conditional phrasing for these latter effects is

important, as WW overall find mixed results on these effects depending on setting. They attribute this at least in part to changes in bid shading strategy in equilibrium by the ZI traders.

Our *BestGuess* implementation of the model meanwhile shows clear benefits of fragmentation in the experimental scenarios. Fragmentation in this implementation generally increases surplus, decreases execution times, and decreases NBBO spreads. The average exchange-level BBO spread is increased, however. LA reduces surplus and exacerbates spreads relative to the fragmented market without LA, but LA also further reduces execution times. The fragmented market with LA outperforms the single-CDA setting for most metrics in most settings. These effects are also more consistent than those reported by WW, with less variation for specific experimental settings.

Comparing the results from our *BestGuess* and *BestGuess+MS* implementations of the model provide a clearer contrast, as we know the only difference between these two models is the implementation of the ZI traders' greedy strategy. The greedy strategy taken from *MarketSim* has the ZI traders base their decisions only on their primary exchange BBO and submit overly aggressive order prices to try to hit the observed price. This results in reduced surplus and increased execution times relative to our *BestGuess* implementation, where traders also utilize the NBBO and submit their orders at the observed market price. The less strategic ZI actions taken in the *BestGuess+MS* implementation also result in greater arbitrage opportunities, as evidenced by higher LA surplus values and more LA transactions.

Overall, the implementation of the greedy strategy for the ZI agents has a marked impact on a number of these takeaways, particularly for the effects of fragmenta-

tion relative to a consolidated market. When the ZI agents submit less aggressive greedy orders and factor the NBBO into their strategy, the presence of fragmentation leads to reduced execution times in each experimental setting and generally increases trader surplus as well. These results suggest that traders acting more strategically helps account for information uncertainty and that market fragmentation has clearer benefits when this is the case. This has parallels with the findings from the Duffin and Cartlidge [66] replication of the Wah and Wellman [190] model. They found that the background traders using the ‘minimally intelligent’ ZIP trading algorithm led to results showing fragmented markets benefit from latency arbitrage. The sensitivity of the qualitative takeaways to the specifics of the greedy ZI strategy in the 2016 model [189] at the very least highlights that this strategy should have been given more than a two-sentence description in the original paper. More broadly, these results show that further exploration by the modeling community is needed to separate the effects of fragmentation and latency arbitrage from the specifics of trader agent implementation.

3.6 CONCLUSION

Wah and Wellman [189] made a valuable contribution to agent-based market modeling by incrementally introducing elements of realistic market structure in order to assess their effects. They note,

As with any simulation model, our results are valid only to the extent our assumptions capture the essence of real-world markets. [189, p. 91]

Replication tests the extent the documented assumptions and results capture the

essence of the *model*.

We find that the specifics of the background trader greedy strategy prove especially important for the overall behavior of the model. We cannot claim that others would necessarily have interpreted the details given by Wah and Wellman [189] the same as we did in our *BestGuess* implementation. We argue that our interpretation better fits the spirit of their description. Specifically, as opposed to the *MarketSim* implementation, the following properties hold in our *BestGuess* implementation of the strategy:

1. The ZI agents use all information to which they have access.
2. The strategy with $\eta = 1$ is equivalent to not using the strategy.

Regardless, the sensitivity of the original study’s qualitative takeaways to this implementation detail limits the range of applicability for those takeaways to real-world markets. A more comprehensive exploration of the parameter space with our implementation could yield further insights into the range of behavior resulting from our assumptions. Further work should also be done to validate the behavior of the model against stylized facts of empirical market behavior, test the framework with different ecologies of traders, and extend the framework to incorporate yet more realistic elements of fragmentation and modern market structure.

Our experience from this replication effort offers takeaways for agent-based modeling and replication more broadly. The first is that the replicator should test the reliability of their replication criteria for the model in question. Some studies have checked this by testing the result distribution for normality in order to assess the appropriateness of a given statistical estimator [151, 156]. An alternative approach,

taken here, is to test the replication model against itself by taking many independent samples of the specified size and checking alignment between samples. This comes with computational tradeoffs, requiring many more runs than the original study. The bootstrap confidence interval methodology we use here to test quantitative alignment proved a useful replication guide. Our implementation of the model with the best quantitative alignment to WW’s ZI and LA surplus metrics under this methodology also had the best qualitative alignment to the original model across the various metrics.

Next, increasing the complexity of the model corresponded with increased difficulty replicating the results. Even in our best replication implementation, we reject quantitative alignment for all model settings with nonzero latency, and we achieve relationally equivalent results for most but not all metrics and settings. The existence of a public codebase from the original study’s authors helped uncover important implementation details of the model, but the codebase failed to replicate the original model. The complexity of the model and codebase made it challenging to determine what is driving the divergence. As the model becomes more complex, the number of design choices increases and the sensitivity of the model to certain choices may also increase. More complex models thereby need even more thorough documentation (and more acutely need to be independently replicated).

Edmonds and Hales provocatively said, “An unreplicated simulation is an untrustworthy simulation” [67, p. 12.2]. This would render most agent-based market models untrustworthy, in line with the SEC’s expressed sentiment [180]. By encouraging a culture of replication [171], market modelers can build more trustworthy models that can then be extended to better capture the complexity of the financial system.

CHAPTER 4

HOW DOES MARKET FRAGMENTATION AFFECT PRICE DYNAMICS? EVIDENCE FROM AN AGENT-BASED MODEL

ABSTRACT

The U.S. equity market is fragmented and has become more so over recent decades. The Securities and Exchange Commission (SEC) has warned that increased fragmentation could decrease price competition among orders, in turn impairing price discovery and creating excessive short-term volatility. However, financial agent-based models (ABMs) still largely represent trading as consolidated, limiting their application for studying how current market structure may shape aggregate behavior. In this study, we extend an existing fragmented-market ABM by adding trading strategies adapted from prior work, using the resulting model to compare price dynamics gener-

ated under consolidated and fragmented configurations. We find that the interaction between market structure and the information acted on by agents is central to how fragmentation affects price formation and aggregate returns.

When agents act only on price quotes from their local exchange, market fragmentation can amplify extreme price moves and concentrate them in time. The resulting returns exhibit heavier tails, stronger volatility clustering, and less persistent linear autocorrelation than the consolidated-market benchmark. When agents instead act based on cross-market price quotes from a consolidated feed, the effects of fragmentation on the generated price dynamics are much smaller, though latency on that feed makes them more visible. Adding a latency arbitrageur to the fragmented configurations has minimal incremental effect. More broadly, the study illustrates how ABMs with explicit market microstructure can be used to compare structural scenarios and clarify the mechanisms linking market organization to aggregate behavior.

4.1 INTRODUCTION

Since... nature is only nearly-decomposable, not completely decomposable, many of the most beautiful regularities of nature will only be approximate regularities. They will fall short of exactness because the properties of the lower-level, higher-frequency subsystems will ‘show through’ faintly into the behavior of the higher-level, lower-frequency systems

—Simon [163].

The number of equity exchanges in the National Market System has doubled

(from 9 to 18¹) since 2005 when the Securities and Exchange Commission (SEC) adopted rules (Regulation NMS) aimed at modernizing and strengthening the regulatory structure of the U.S. equity market. In addition to the national exchanges, an increasing portion of trading in NMS stocks occurs on dozens of alternative trading systems (ATSs, or ‘dark pools’) or over-the-counter (OTC) at hundreds of other institutions [178]. The SEC has warned that fragmentation could decrease price competition among orders, in turn impairing price discovery and creating excessive short-term volatility [53]. Because fragmentation is an inherent feature of the NMS, isolating its effects empirically is challenging [140, 181]. Financial agent-based models (ABMs) provide a natural complement to the empirical approach, allowing for controlled experiments in a computational laboratory [74]. Most ABMs ignore the fragmented structure of the market, however, instead representing a single exchange or no explicit market mechanism at all. This may contribute to concerns the SEC has recently expressed about the ability of these models to inform its policymaking [180] (p.328).

A primary motivation for the development of financial ABMs has been to endogenously reproduce aggregate ‘stylized facts’—descriptions of behavior widely accepted to be robust features of the empirical system, as discussed in Chapter 2. In particular, the stylized facts most commonly used to empirically benchmark financial ABMs describe the properties of price returns [42, 44, 54], providing a check on the aggregate price process generated by the modeled system. In their review of ABMs in economics and finance, Axtell and Farmer argue that “models that cannot reproduce qualitative aggregate phenomena like clustered volatility, heavy-tailed return distri-

¹The 18th stock exchange, the Texas Stock Exchange (TXSE), began trading in July 2026 [40, 170].

butions, and log price time series with little autocorrelation should be viewed with skepticism” [14, p. 43]. The majority of financial ABM studies implicitly, or even explicitly [127], assume that representing multiple trading venues in a market is an unnecessary complicating factor for producing these aggregate phenomena.

Aggregation and abstraction are natural (and indeed necessary) parts of the modeling process [164, 165]. Simon [165] argues that the nearly-decomposable, hierarchical² organization of many complex systems contributes to our ability to understand, model, and even identify these systems. Much of the aggregate behavior of such a system can be understood from its organization and the details of its high-level subsystems while abstracting away the detail of lower-level components. At the same time, the properties of the lower-level subsystems will ‘show through’, since the high-level systems are not truly separate from the underlying structure [163]. The strength and nature of the inter-system connections determine how well our approximations capture the mechanisms behind observed dynamics [163].

In this study, we use an ABM to examine how fragmented market structure ‘shows through’ into aggregate price dynamics. We begin from a replication of Wah and Wellman’s 2016 model [189], which, to our knowledge, was the first ABM to represent communication latency and latency arbitrage in a fragmented market. We use our replication from Chapter 3 as the baseline for this implementation. We then adapt the trading strategies from Vyetrenko et al. [187] to the fragmented market environment of the Wah and Wellman model. With the extension, we update the model’s formulation of time to simulate standard, 6.5-hour trading days measured at nanosecond granularity. This allows us to more directly compare our modeled latency settings

²Hierarchy here refers to nested subsystems rather than subordinated authority [165].

and simulated results to empirical findings from the NMS. We demonstrate that the adopted agent configurations produce qualitatively similar results to Vyetenko et al. in the consolidated-market setting of the Wah and Wellman model, although we also identify some differences and discuss possible causes. With the resulting model, we begin from a consolidated market with no latency before incrementally adding another exchange, communication latency, and a latency arbitrageur. Across these configurations of the model, we consider two extremes of how agents access price quotes in a fragmented market: either they only access quotes from their local exchange or they have access to cross-market quotes from a consolidated data feed with a small amount of latency.

Using controlled experiments simulated with paired random seeds across configurations, we measure how structural changes cause prices to diverge relative to those produced under the consolidated-market baseline. Simulated prices differ due to changes in each of the structural elements varied across our experiments, but the degree of these differences is not uniform. Fragmentation alters price dynamics most when traders only have access to price quotes from their local exchange. The resulting returns exhibit heavier tails, increased volatility clustering, and less persistent linear autocorrelation relative to the consolidated baseline. Fragmentation of order flow alone has more limited effects when traders are assumed to have immediate access to consolidated cross-market quotes. Introducing latency to the consolidated data feed produces slightly larger effects, but also not to the level seen from fragmenting access to information.

We give more background on market fragmentation and market models in the next section (Section 4.2) before giving an overview of the model setup (Section 4.3). We

then detail our integration of the trading strategies from Vyetrenko et al. [187] into the model, showing qualitative alignment with most of their reported results (Section 4.4.1). We detail our fragmentation experiment results in Section 4.5. Finally, we discuss implications, limitations, and possible future extensions of our results in Section 4.6.

4.2 BACKGROUND

Trading for a stock in the modern NMS is distributed across many venues, as is the information establishing the best displayed prices for the stock at any point in time. A stock is listed at a primary exchange, but it can be bought or sold at any of the other exchanges or off-exchange venues [33]. Trades occur by matching orders to buy and sell the stock, and the best displayed prices are established by the best offers to buy or sell the stock displayed at the different exchanges. The exchanges are required to disseminate top-of-book quotations and associated sizes to the Securities Information Processors (SIPs), which act as data consolidators for the market. By contrast, off-exchange venues must report trades but generally do not disseminate quotations publicly to the market [79]. This has led to the common characterization of exchange trading as ‘lit’ and most off-exchange trading as ‘dark’. We focus primarily on the lit market here, as the model implements a simplified version of the high-level system of stock exchanges, SIPs, and connecting data feeds.

During regular continuous trading, each stock exchange in the NMS today operates an electronic limit order book (LOB) implementing a continuous double auction (CDA) as its primary market mechanism. A LOB tracks all orders to buy and sell

the stock at the exchange, facilitating trades when an incoming buy or sell order matches against a previous order that is still active on the LOB. There are various order types with different rules, the most fundamental types being the limit order and the market order. A limit order specifies a commitment to buy (or sell) up to some quantity q of the security at a maximum (minimum) limit price p . An incoming buy limit order is marketable if it is priced at or above the price of the best resting ask price. Similarly, an incoming sell limit order is marketable if it is priced at or below the best resting bid price. A marketable order executes against resting orders on the opposite side of the book. Active limit orders in the LOB are prioritized via some ordering specified by the rules of the exchange. For example, price-time priority means orders with better prices are executed first, and orders with the same price are ordered by the time they were received (first-in-first-out). If the incoming limit order is not marketable or was only partially filled, it becomes an active limit order on the book until it is filled or canceled. A market order specifies a commitment to immediately buy (or sell) some quantity q of the security at the best available price. A market order for q shares will match against active orders on the opposite side until all q shares have been exchanged or no more orders are available on the opposing side of the book.

Active limit orders provide liquidity to the market and establish the best prices displayed at a point in time. The buy limit order with the highest price establishes the best bid at the exchange, while the sell limit order with the lowest price is the best offer (ask). The exchanges publish this best bid and offer (BBO) information to the market, through the SIP or directly via proprietary data feeds. Because orders are distributed across multiple exchanges, the best local bid or ask at one exchange

will not necessarily be the best available bid or ask across the market as a whole. The SIP consolidates exchange-level quotations into the national best bid and offer (NBBO), which provides the public consolidated benchmark for the best displayed bid and ask. Alternatively, a trader can subscribe directly to the proprietary feeds from the exchanges and construct their view of the cross-market best bid and ask, often with lower latency than the SIP feed [141].

Multiple stock exchanges have been simultaneously operational in the U.S. dating back to the 1800s [132]. In 1975, Congress mandated the establishment of a national market system, tasking the SEC with linking together these individual markets [53]. In creating and maintaining the NMS, the SEC has sought to avoid the extremes of a monopolistic consolidated market and a fragmented market in which orders are uncompetitively split across venues [53]. Toward these ends, the SEC in 2005 adopted a set of rules known as ‘Regulation NMS’ aimed at increasing market competition, as referenced in our introduction.

A major feature of Regulation NMS is the Order Protection Rule (Rule 611), which was introduced to promote faster, automated trading and eliminate competitive advantages that prior rules gave slower, manual markets [53, 181]. The rule requires trading centers to maintain policies and procedures reasonably designed to prevent trade-throughs of protected quotations. A trade-through occurs when an order is executed at a price inferior to a protected quotation displayed elsewhere in the market. Preventing trade-throughs therefore requires a trading center to have timely information about whether a better-priced protected quotation is displayed elsewhere in the system.

Analogous rules prior to Regulation NMS were less stringent and did not distin-

guish between quotations on manual versus automated markets. At the time Regulation NMS was adopted, the SEC estimated that 1 out of every 40 trades in both Nasdaq and NYSE stocks represented a ‘significant trade-through of a displayed quotation’ [53] (p.26). Manual trading floors operated by some exchanges (e.g., NYSE) often had latency greater than 15 seconds. This made it slow and costly for electronic trading systems to ensure an order was executed at a fair price compared to these slower markets, thus limiting their ability to compete for market share [9, 53]. By comparison, fully electronic markets such as Nasdaq had very low latency, generally less than a second [53]. As a result, other automated venues were better able to compete for order flow of Nasdaq-listed stocks. With the adoption of Regulation NMS, NYSE’s executed market share for NYSE-listed stocks decreased from 79% in 2005 to 29% in 2009 [181]. The three trading venues operated by Nasdaq had a market share of 52% for Nasdaq-listed stocks in 2005, which similarly decreased to 33% by 2009 [181]. With the changes in incentives, NYSE changed its trading system, first into a hybrid floor and electronic limit order book (LOB) system, and then into a fully electronic LOB [128].

The exchanges and SIPs are now all largely automated and facilitate the fast communication of information across a fragmented market system. This communication is not instantaneous, however. Each of these electronic systems has a physical location—generally, a data center where the system operates. The data centers housing the automated matching engines for the stock exchanges in the NMS are split across four locations in north-eastern New Jersey: Mahwah, Carteret, Secaucus, and Weehawken [184]. Information cannot travel faster than the speed of light, which therefore sets a lower bound on how long it takes for a message to travel from one

data center to another [8, 69]. In addition to geographic latency, technological latency is introduced at each step. The structure of the market therefore splits liquidity across venues and can create different local views of the market depending on where a participant in the market is located and to which data feed(s) it is subscribed.

4.2.1 FINANCIAL ABMs

As highlighted above, the NMS is highly fragmented and has been throughout recent memory [53, 132]. Empirical studies of the effects of fragmentation must therefore either compare trading data from fragmented markets to data from some other, consolidated market, or they must compare data between different assets or time periods with differing degrees of fragmentation. In any of these scenarios, separating the effects due to levels of fragmentation from differences in the underlying asset, time period, or market can be quite challenging [140]. ABMs provide an alternative (albeit complementary) approach by providing a means of directly testing our understanding of the mechanisms underlying observed phenomena. Per Epstein’s motto of generative social science, “If you didn’t grow it, you didn’t explain its emergence” [74] (p.43).

A primary motivation for early financial ABMs was to generate the stylized facts of returns endogenously from the interaction of heterogeneous, boundedly-rational traders [57, 99, 120, 144, 150]. These efforts emphasized the ecology of trading agents and how specific strategies, or mixtures of strategies, can give rise to the stylized facts [57, 101, 119]. LeBaron [111] in 2001 argued that the representation and structure of trading agents is the most important design decision, ahead of the specific trading or price-formation mechanism. Most early agent-based market models

addressed price formation via aggregate excess demand or through an entity that computes a market-clearing price [57, 111, 118, 150]. Chen, Chang, and Du [44] contend that these models are primarily geared towards explaining regularities of low-frequency data.

An alternative route is to explicitly model a trading mechanism that mirrors real markets [45]—generally, a CDA market similar to those operated by exchanges in the NMS [42]. LeBaron notes that this approach “is most appealing for modeling high-frequency data” [111] (p.6). Prior work³ has demonstrated that the structure of the trading mechanism can itself be a primary determinant of statistical properties of prices [26, 83], even absent strategic agent behavior [76, 149, 167]. The Farmer et al. ‘zero-intelligence’ model [76, 167] and models in its lineage [25, 149] specify the rate at which traders submit orders that will cross the spread, resulting in a transaction, versus being shaded some distance away from the opposite side of the book, thereby resulting in a new limit order being added to the LOB. Thus, even with a zero-intelligence (ZI) formulation of agent behavior in a CDA market, modeled agent strategies are often responsive to the bid, ask, and spread of the LOB at a given point in time.

In a pair of studies, Wah and Wellman [189, 190] extended the representation of detailed market structure to introduce an additional CDA exchange, a SIP with communication latency, and a latency arbitrageur (LA) agent. To our knowledge, they were the first to consider the effects of these structural elements of fragmentation in an ABM. Their initial 2013 version of the model featured a simple, fixed strategy for its background traders, similar to the ‘zero-intelligence constrained’ (ZIC) strategy

³See the reviews from Chakraborti et al. [41, 42] and Chen, Chang, and Du [44] for a more complete review of financial ABMs in relation to aggregate stylized facts.

introduced by Gode and Sunder [86]. Their ZIC traders each act in the market exactly once throughout the simulation, placing an order randomly within a certain range independent of the state of the best prices in the market at that time. The order is submitted to the trader’s local exchange unless it could execute immediately at another exchange with a better price according to the SIP NBBO.

Wah and Wellman extended their original model in 2016 [189]. The updated model included a more sophisticated experimental design and a more complex background-trading strategy. Traders in the updated model calculate a private valuation of the asset based on the exogenous fundamental value and their private marginal benefit from being long or short some quantity of the asset. The traders seek to achieve surplus value by shading their order price some amount away from their private valuation. Market quotes are used by the trader to evaluate whether a more aggressively-priced order could immediately achieve some amount of surplus. In Chapter 3, we reported sensitivity of the model’s results to the specifics of this ‘greedy’ strategy extension, including whether traders use market-wide or exchange-specific quotes to update their price. Wah and Wellman’s studies [189, 190] and subsequent replications (including Duffin and Cartledge [66] and our work in Chapter 3) primarily focused on the effects of fragmentation on metrics of allocative efficiency and liquidity. The 2013 study [190] additionally considered the effects of fragmentation on price volatility and price discovery. Wah and Wellman [190] reported that fragmentation reduced volatility and harmed price discovery, although the latter results were not shown in the paper.

A few other studies have considered additional scenarios of market fragmentation [174, 175, 183], building from the foundation set by Wah and Wellman’s framework. Tivnan et al. [175] considered different mechanisms of coupling exchanges in a

fragmented market, finding that the SIP contributed most to efficient price discovery. Tivnan et al. [174] and Van Oort et al. [183] each introduced additional features of the NMS, such as message latency, additional order types, circuit breakers, and, in the case of Tivnan et al., a dark pool. They report the ability of their models to reproduce aggregate stylized facts of price returns with these additional structural features, but they do not consider the incremental effects of the structural elements on those measures. Studies have also simulated fragmented markets with one dark pool alongside one lit exchange [133, 134, 135]. Mizuta et al. [133] report impacts to volatility and kurtosis from the level of trading at the dark pool versus on the lit exchange.

Other models have applied features of Wah and Wellman’s model [189, 190] to scenarios of trading in a consolidated market. Byrd et al. [38] tested the effects of latency on trader outcomes in a consolidated CDA market, finding that a trader’s profitability depended more on its latency relative to other traders than its absolute magnitude of latency. Byrd et al. incorporated this variable latency setup and an extended version of Wah and Wellman’s exogenous fundamental-value process into the open-source ABIDES market simulation platform [35, 36, 38]. The platform has since been used in a number of simulation studies (e.g., [37, 121, 187]). Vyetrenko et al. [187] used the platform to demonstrate the effectiveness of agent-based modeling approaches for evaluating trading strategies and ‘what if’ scenarios in a simulated trading environment. They propose a set of empirical regularities summarizing behavior across multiple levels of LOB architecture, testing a subset of these metrics to evaluate configurations of traders in the ABIDES platform.

The incremental effects of the structural features of fragmentation on price dy-

namics have not been as directly explored in a computational setting. Perhaps the most similar study to ours in approach if not in exact subject is Bottazzi, Dosi, and Rebesco [26], which compared the dynamics simulated from consistent agent settings under different market mechanisms: a Walrasian auction, a batch auction, and a LOB mechanism. Bottazzi et al. found that “some basic dynamic properties of the time series of price, volumes and returns, like skewness, kurtosis and autocorrelation profiles, are largely shaped by the specific architectures of trading mechanisms, in ways largely independent from differences in traders ecology” [26] (p.226). They report negative skewness and high kurtosis for the batch auction, while these metrics were roughly zero for returns from their order book simulations. They compare these results to daily skewness and kurtosis measures from LSE, with close and opening auctions corresponding to their batch auction scenario and open-to-close behavior analogous to their LOB scenario. Bottazzi et al. also note that market institutions shape the information agents access—a key insight we similarly highlight for our study.

4.3 MODEL OVERVIEW

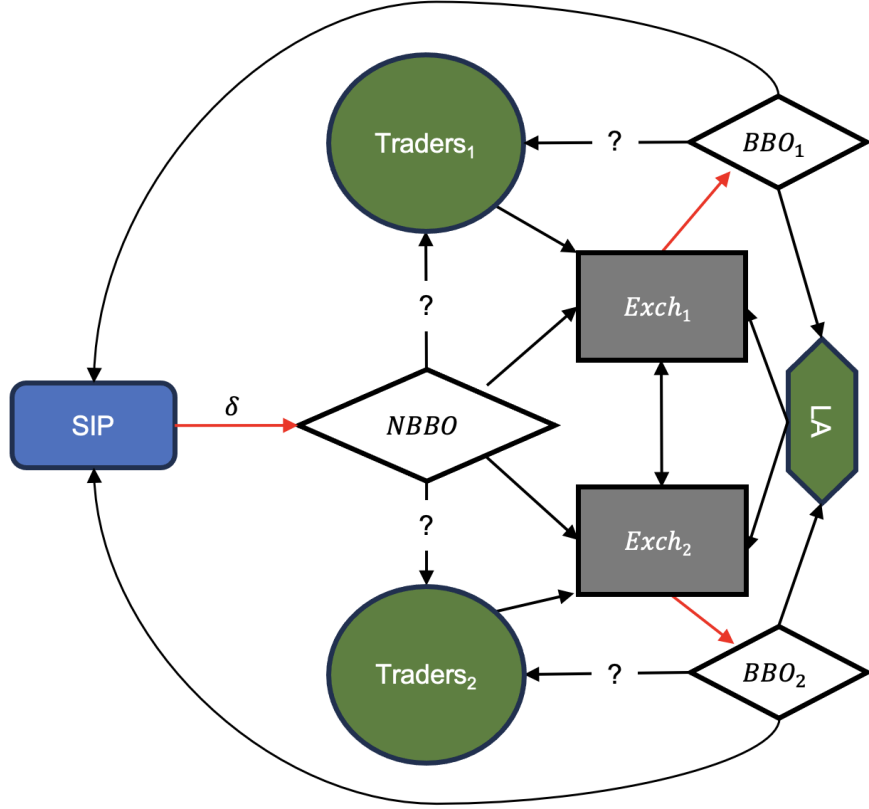


Figure 4.1: High-level overview of the modeled market structure with two exchanges and one latency arbitrageur (LA). Black arrows denote communication between entities, whereas red arrows denote internal processing within an entity. Each exchange $Exch_i$ publishes its local best bid and offer feed, BBO_i . The securities information processor (SIP) consolidates these exchange-level BBO feeds and, after latency $\delta \geq 0$, publishes the national best bid and offer (NBBO). Each trader is assigned to a single exchange and submits all of its orders there. The question marks denote the experimental information setting: when making trading decisions, traders observe either only the NBBO or only the BBO of their assigned exchange. After receiving an order, an exchange may route it to the other exchange if the current NBBO indicates that a better execution price is available there. The LA agent observes all exchange BBO feeds without latency, identifies the best prices available across exchanges, and uses this information to detect and exploit arbitrage opportunities.

In this section, we give an overview of our general model structure. This structure is shared across all our experiments and is diagrammed for a configuration with two

exchanges in Fig. 4.1. We build our fragmented market model from the framework introduced by Wah and Wellman [189, 190]. We specifically extend the 2016 version of Wah and Wellman’s model, beginning from our replication detailed in Chapter 3. We provide a more detailed description of our extension following the ODD (Overview, Design concepts, Details) protocol [90, 92] in Appendix C.1 to facilitate future replication, critique, and extension of the model.

The modeled market features a single fungible stock traded on $N_{\text{exch}} \geq 1$ exchanges, each running a CDA limit order book. All orders are limit orders, with positive integer prices. All orders in the original model [189] are assumed to have fixed quantity $q = 1$, but we generalize this to allow order size to be any positive integer. This is needed to accommodate the variable-size strategies introduced in Section 4.4.1. Each exchange publishes the best bid and ask prices in a BBO feed, as well as the total quantities available at the bid and ask.

Time is discrete, with $t \in \{1, \dots, T\}$, where T is the simulation length. A scheduler maintains a queue of events ordered by time and insertion order (first-in, first-out). The SIP consolidates the exchange BBOs to construct a national best bid and offer (NBBO). The SIP’s NBBO calculation occurs with some latency $\delta \geq 0$, such that an update received at time t is reflected in the updated NBBO at time $t + \delta$. All other events are assumed to be ‘instantaneous’, kicking off a specified sequence of sub-events that complete without going through the scheduler or incrementing the time. Thus, the only communication latency in the model comes from the calculation of the SIP NBBO.

In the model, each exchange uses the SIP NBBO to determine whether an incoming order should be rerouted to avoid an immediate trade-through. When an

exchange receives an order, it first performs logic to determine whether the order would immediately execute at a superior price at another exchange in the market, in which case it reroutes the order to that exchange⁴. If any portion of an incoming order cannot match against resting orders at the exchange nor at any other exchange according to the NBBO, it is added to the exchange’s LOB as a resting order. The trader that submitted the order can cancel it at any time.

To illustrate how fragmentation can affect trading dynamics, consider the scenario detailed in Fig. 4.2. The same set of resting orders are shown in a consolidated LOB and spread out between two exchange LOBs. An incoming order is submitted at time t to buy the stock at price \$101. In the consolidated market, the incoming order will match against the resting sell order at that price, resulting in a trade. In the fragmented market, the order—originally routed to Exchange 1—must be rerouted in order to prevent a trade-through and, in this case, a locked market (where the best bid and best ask are equal in price). If the NBBO is stale when Exchange 1 receives the incoming order, the exchange might not be rerouted, instead resting on Exchange 1’s LOB and creating a locked market.

The tables in Fig. 4.2 highlight the different possible views of market information under each setting. The SIP NBBO is equivalent to the consolidated market BBO if there is no latency or if sufficient time has passed between events for the SIP to catch up. Prior to the trade, a trader looking at only one of the exchange BBOs will have a different view of the bid, ask, and spread than if it were able to see the

⁴In the original model, Wah and Wellman [189, 190] describe the background traders as performing their own routing logic based on their assigned exchange’s BBO and the SIP NBBO. Because there is no latency between an agent submitting an order and the order being processed by the exchange, this logic is equivalent to having the exchange do the routing logic. We confirm that this change makes no impact to the model, producing identical results to original logic when settings and random seeds are kept constant.

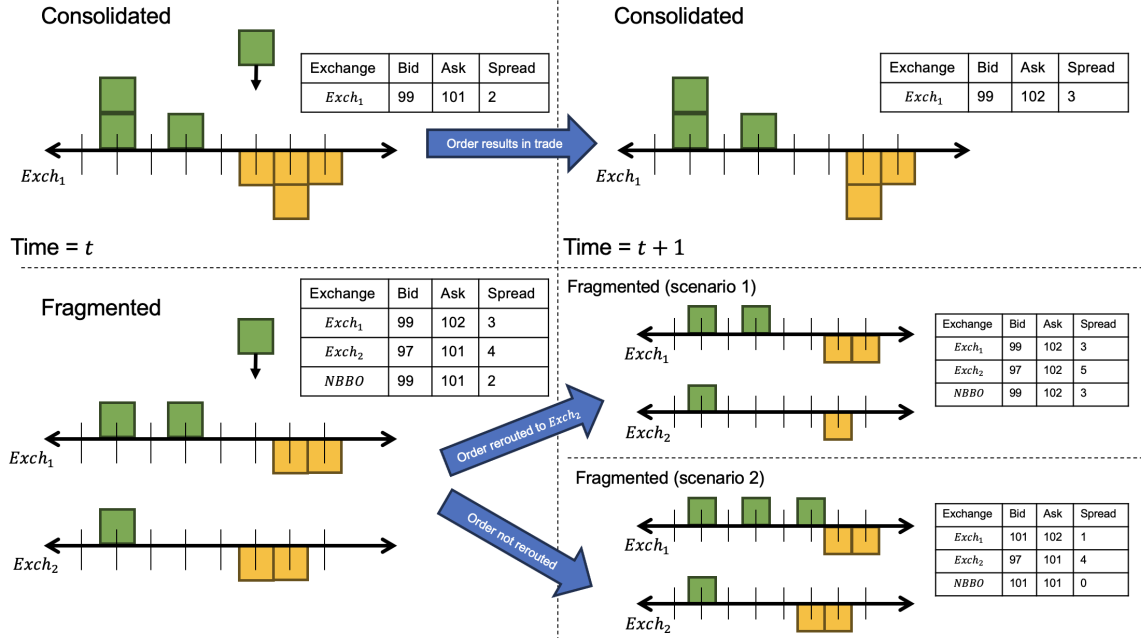


Figure 4.2: Example comparing a scenario of orders placed in a consolidated versus fragmented market. The same set of buy limit orders (in green) and sell limit orders (in yellow) are placed in a market with a single exchange or a market with two exchanges. Each exchange operates a limit order book with a first-in-first-out queue at each price point (i.e., price-time priority). For simplicity, assume all orders have quantity $q = 1$. The best buy price available at an exchange is the ‘bid’ and the best sell price available is the ‘ask’. The national best bid and offer (NBBO) tracks the overall best bid and ask prices across the market. The green order with an arrow is an incoming buy order at price \$101. In the consolidated market, the order will match against the sell order at the ask price (\$101). In the fragmented scenario, the order must be rerouted to exchange Exch₂ for the trade to occur. Otherwise the order will be added to Exch₁’s book, resulting in a locked market with bid = ask.

consolidated view. After the trade (assuming the order was appropriately rerouted), Exchange 1's BBO happens to match the consolidated market BBO, but Exchange 2's BBO does not. A trader determining its next trading decision based on market information may therefore take a different action depending on what data feed(s) it has access to. Under the alternative scenario where the order was not rerouted and resulted in a locked market, the consolidated-feed view of the market (once updated) will show a spread of zero, which cannot occur in the single-exchange market. From the above, we see the following ways in which fragmentation can affect trading: by splitting orders across exchanges; by creating the possibility of a locked or crossed market; and by fragmenting information that must then be efficiently consolidated if traders are to have an accurate view of market quotes.

If the market becomes crossed—with the best available bid price greater than the best ask price available at that moment—it creates an arbitrage opportunity for a trader that can recognize and act on it. The model assumes an ‘infinitely fast’ latency arbitrageur (LA) agent that exclusively acts on such scenarios. The LA agent subscribes to the BBO feeds from each exchange in the market, constructing its own ‘direct best bid and offer’ (DBBO) from the best bid and ask prices available across the exchanges. Let BID_t^{DBBO} be the direct best bid and ASK_t^{DBBO} be the direct best ask at time t . Also, let α_{LA} be some small threshold margin the LA agent uses to determine whether an arbitrage opportunity is worth acting on. Whenever $BID_t^{\text{DBBO}} > (1 + \alpha_{\text{LA}})ASK_t^{\text{DBBO}}$, the LA agent submits orders to buy the stock at the best ask and sell at the best bid, resulting in an immediate profit. We assume that the LA agent can bypass the exchange's rerouting logic.

All other traders in the model are referred to generically as ‘background traders’.

These traders generate the order flow that drive all other events in the model. We detail the background trading strategies in the following sections, first for the agents used by Wah and Wellman [189] and then for the trading strategies taken from Vyetenko et al. [187]. Each background trader is assigned an exchange where they submit all of their orders. We will refer to that exchange as the trader’s ‘local’ exchange going forward. An order may then be rerouted by the exchange based on the NBBO, as detailed above.

The model assumes the stock has an exogenous fundamental-value time series that evolves stochastically over time, which some trader agents use as part of their strategy. Price series for the stock, however, emerge endogenously from the interaction of orders on the exchange(s). An Observer entity collects (without latency) the quote and trade data generated over the course of the simulation. These data are then used to construct the price, return, and quote series used in our subsequent analyses.

4.3.1 INFORMATION REGIMES

We consider two possible configurations of how the background trader agents receive market information. First, under the ‘NBBO information regime’, all agents subscribe to the NBBO. Under this scenario, background agents have a shared (albeit possibly delayed) view of the market at each point in time. This setup is analogous to the fragmented market behaving like a single virtual exchange with multiple points of entry [140]. The alternative is the ‘BBO information regime’, under which each background agent only has access to its local exchange BBO. A background agent’s trading decision will then be based solely on the price information visible on its local exchange, with no direct visibility to quotes displayed elsewhere in the market. This

scenario represents an alternative extreme to the single virtual exchange, where instead information is split across venues. Real-world markets are more likely a complex mixture of these two options, with some traders relying on the SIP while some (like the LA agent) subscribe to the direct BBO feeds from all exchanges. However, roughly half of trading volume in NMS stocks now occurs off-exchange at venues that do not display liquidity to the market [`cboe_exchanges_inc_cboe_nodate`]. Hence, fragmented information is relevant to today’s market.

4.4 ALIGNMENT AND EXTENSION

We begin from the replication of the 2016 Wah and Wellman model [189], which we refer to as ‘WW’ going forward. We then extend the model to support the additional analyses targeted in our study and to incorporate an updated ecology of background trader strategies, as detailed in Section 4.4.1. We confirm our extended version of the model reproduces the results reported in Chapter 3, verifying that these extensions have no impact when running with the settings specified by Wah and Wellman [189]. This verification exercise is reported in full in Appendix C.2.

Independent replication and extension are vital steps in developing a cumulative science. Our independent replication effort documented in Chapter 3 provided implementation details and nuance not found in the original study. In particular, Wah and Wellman’s [189] overall takeaways on the negative effects of fragmentation and latency arbitrage were found to be sensitive to specific elements of how the background traders choose to place aggressive orders. An alternative implementation of this part of the trading strategy that is still consistent with the prose of the original

study produces contrary results. Under the alternative assumptions, trader welfare and execution times were helped by the introduction of an additional exchange in the market, even with latency and latency arbitrage. Duffin and Cartlidge [66] report similar effects for Wah and Wellman’s original 2013 model [190]. Duffin and Cartlidge found sensitivity of the model to the specific shading parameter used by the ‘zero intelligence constrained’ (ZIC) traders and report contrary results when the traders were updated to use zero intelligence plus (ZIP) strategies [51].

Extension allows for the development of more complex models by building on the foundations others have established and learning from the effects of their specific assumptions. Wah and Wellman [189, 190] introduced a framework by which to move from a single CDA exchange to a fragmented market with latency and latency arbitrage. The findings reported by Wah and Wellman [189, 190], Duffin and Cartlidge [66], and our replication in Chapter 3 provide a thorough understanding of the incremental effects of these additions for the model’s assumptions, but only for the welfare and liquidity metrics those studies focused on.

In the context of our planned experiments, we ultimately decide the WW model should first be extended, with the goal of improving the relevance of the modeled behavior to the referent system. This decision is based on the following considerations. First, as documented in Chapter 3, the effects of fragmentation in the original model are sensitive to specific features of the background trading strategy that were previously undisclosed by Wah and Wellman [189]. Agent behavior is generally important (and perhaps of primary importance [111]) to the results of an ABM, but we should be careful about sensitivity of the model to artifacts that are not intended to be key elements of our hypothesis.

Second, the original model has an abstract formulation of time, complicating our ability to relate our findings to empirical detail of the real-world NMS. Time in WW is modeled as discrete, taking positive integer values $1 \leq t \leq T$. The simulation lengths considered by Wah and Wellman for their three environmental parameter sets are $T \in \{15000, 10000, 5000\}$. SIP latency δ is set to some fixed number of timesteps, with δ ranging from 0 to 900 depending on the experiment. In the prose of their study, Wah and Wellman suggest that δ is analogous to the latency exhibited by the SIP in the real-world market. Average SIP latency in the NMS has decreased from multiple milliseconds in 2010 to around 10–20 microseconds in the first quarter of 2026⁵. If the SIP latency from the original model is on a similar timescale (microseconds or milliseconds), the total simulation length $T \leq 15000$ is analogous to a very small portion of a standard 6.5-hour trading day; perhaps multiple seconds or minutes.

The background traders base their order prices on estimates of what the fundamental value will be at the end of the simulation. These traders activate every 20 or 200 time steps on average, depending on the environment. This means that the supposedly ‘investor’ traders in the model can act multiple times before the SIP has the chance to update, and they calibrate their trading strategy for gains over a short intraday timescale. Based on these nuances of how time is represented in the original model, it is not immediately clear how to connect the model’s scenarios to time and latency in the NMS.

Finally, while the WW model incorporates additional structural detail based on the NMS, we are not aware of any attempts to test its simulated behavior against empirical behavior of real-world markets. We therefore tested the single-market con-

⁵<https://www.utpplan.com/metrics> and <https://www.ctaplan.com/metrics>

figurations of the WW model for evidence of heavy tails, volatility clustering, volume/volatility clustering, and lack of linear autocorrelation—properties of returns for which researchers have found repeated empirical support [54, 41], including in fragmented markets, as documented in Chapter 2. We summarize the results of this exercise below and detail the results more fully in Appendix C.2.

As just noted, the abstract definition of time in the model complicates direct empirical comparison. We also found the simulated price series exhibits much larger swings at the beginning and end of the simulation than occur throughout the rest of a given run. These outlier returns due to edge effects were found to materially affect some of the benchmarking metrics, particularly heavy tails and volatility clustering. Persistently negative (albeit weak) autocorrelation and weak volatility clustering in the simulated returns are inconsistent with empirical results (see Chapter 2). The simulated return distributions exhibit very large kurtosis values when the opening and closing periods of the simulation were included but approximately normal kurtosis levels were found for $\Delta t > 20$. Note that this timescale is shorter than the SIP update times considered for scenarios where latency is nonzero. Based on these findings, we elect to extend the model with the following primary goals: incorporate a more realistic definition of time and improve the performance of the model against the empirical benchmarks.

4.4.1 WW+GETREAL

We turn to the study by Vyetrenko et al. [187] to update our background trader and environmental settings. The experiments detailed by Vyetrenko et al. address many of our current concerns about the canonical WW model. They simulate full

trading days, thus providing a more realistic formulation of time. This is particularly important: as demonstrated in Chapter 2, at least some of the empirical properties of returns we focus on here are directly related to how time is measured. Vyetenko et al. document some shortcomings of their simulated returns and quote count metrics, but they demonstrate better evidence of heavy tails and volatility clustering than we observed for the WW model. They also provide evidence of an intraday activity pattern similar to that documented in empirical equity markets [103, 187, 197]. Vyetenko et al. developed their experiments using an open-source simulation platform—ABIDES [36]—thereby increasing our likelihood of successfully replicating their results. Byrd [35] also drew inspiration from the background trading strategy developed by Wah, Wright, and Wellman [191] and Wah and Wellman [189] when developing ABIDES, and the codebase includes a version⁶ of the background trading strategy used in the WW model. This makes the integration a natural fit for our extensions.

We replace the background traders from WW with trading strategies based on the configurations tested by Vyetenko et al. [187]. We refer to the extended model as ‘WW+GetReal’, after the title of their study. Vyetenko et al. report results for four different configurations in their experimental results—two different ecologies of agent strategies with two different methods of simulating an exogenous fundamental-value series. We implement the two agent configurations and the randomly-generated fundamental-value process. We do not replicate their experiments using historical stock price data as the fundamental value. Vyetenko et al. argue that using historical data for the fundamental value produces behavior more in line with empirical results.

⁶<https://github.com/abides-sim/abides/blob/master/agent/ZeroIntelligenceAgent.py>

Our main reasons for not replicating those experiments at this time are the following: First, we do not have the historical stock data used by Vyetenko et al., making it unlikely we would be able to closely replicate their results. Second, their stock data were observed from stocks in the NMS, which could therefore introduce exogenous effects of fragmentation into our experiments, even when running for the consolidated market setting. We therefore leave experiments using historical fundamental-value series out of scope at this time.

We assess qualitative alignment with Vyetenko et al. [187] by reproducing as closely as possible the figures they use to empirically benchmark their results. One caveat is that while we aim to faithfully implement their trading strategies, our market environment is not identical to the ABIDES platform, even in the single-exchange configuration of our market. The main difference we are aware of between the ABIDES platform and our market environment is that ABIDES features variable communication latency between the agents and the exchange⁷. Changes to the information traders act on can alter aggregate behavior (indeed, this is a core argument of this study). However, we limit ourselves here to the structural features of fragmentation from the WW model. We therefore only adopt the core logic of the trading strategies used in the GetReal study and not the full architecture of the ABIDES platform. This difference in trading environment could in turn lead to differences between our results and theirs, as discussed later in this section.

We use the details from the ABIDES platform [36] to implement the GetReal strategies, as well as the parameters that were likely used for the GetReal experimen-

⁷See Byrd et al. [38] for an in-depth exploration of the effects of these latency settings in the ABIDES platform.

tal results. We specifically use the default parameters from the `random_fund_value`⁸ and `random_fund_diverse`⁹ configuration files, which specify agent settings matching the high-level details provided by Vyetrenko et al. [187].

The ABIDES platform treats time as discrete, similar to WW, but at a much finer scale of resolution. Each timestep t is now directly analogous to a nanosecond, and the simulation is run for a standard 6.5-hour trading day ($T = 23400 \times 10^9$). We implement the following trading strategies: noise agents, value agents, market maker agents, and momentum agents. We integrate these strategies into our core model detailed in Section 4.3. Agents of a given strategy type are arbitrarily assigned local exchanges, with equal portions assigned to each exchange (or as nearly as possible). Exchange assignments are determined using a round-robin strategy.

Similar to WW, when a trader is activated it first schedules its next activation and cancels any of its outstanding orders from prior turns. The exceptions are the noise traders, which each activate once per simulation. The trading strategies are introduced below and detailed more fully in our ODD (Appendix C.1).

Noise agents

Vyetrenko et al. [187] describe the noise agents as simulating the actions of retail traders. Each noise agent is activated once per simulation. The noise agent activation times are drawn for a U -quadratic distribution over the interval $[1, T]$. Upon activation, the agent randomly decides to buy or sell and submits an order. In their paper, Vyetrenko et al. describe these agents as submitting market orders. However,

⁸https://github.com/abides-sim/abides/blob/master/config/random_fund_value.py

⁹https://github.com/abides-sim/abides/blob/master/config/random_fund_diverse.py

in the code¹⁰ the agents submit limit orders at the bid price if selling or at the ask price if buying. If the respective bid or ask price at the exchange is equivalent to the agent's latest observation and there is sufficient liquidity at that price to fill the order, this is equivalent to submitting a market order. However, if the agent misses the expected bid or ask due to latency, or if there is insufficient liquidity at that price to fill the order, the remaining quantity will result in a new limit order resting on the book.

We model the noise agent as randomly submitting a limit order at the bid if selling or the ask if buying. The agent will skip its turn if the opposite side of the book is currently empty. Each noise agent's order size is drawn as a random integer in the interval $[20, 50)$.

Value agents

The value agents enact a strategy that tracks the fundamental value of the stock, buying when the agent believes the stock is undervalued and selling when it believes the stock is overvalued. The background trading strategy used in WW also acts based on the fundamental value of the stock. However, in the WW strategy, the agents have a perfect view of the current fundamental value and know the underlying generating process parameters for the fundamental value, which are then used to produce an expectation of the terminal fundamental value at time T . The GetReal value agents have only a noisy view of the current fundamental value, and they do not know the underlying parameters for the fundamental value's random walk.

The value agent therefore constructs an evolving estimate of the fundamental-

¹⁰<https://github.com/abides-sim/abides/blob/master/agent/NoiseAgent.py#L114>

value process and the terminal value. It uses this estimate to determine whether to buy or sell based on the current midpoint for the stock (or it randomly chooses to buy or sell if either best bid or best ask are missing). Some percentage of the time, the trader will act ‘aggressively’ and cross the spread with its order price. Otherwise, it shades its order some distance away from the bid if selling or the ask if buying. Each value agent has a fixed random order size, drawn from the interval $[20, 50)$ when the agent is initialized. Value agents are activated according to a Poisson process, with individual interarrival times drawn from an exponential distribution with rate λ_v . However, each value agent’s first arrival is randomly drawn from the first 100 timesteps (i.e., nanoseconds) of the simulation.

Market maker agents

The market maker agent implements a simple strategy to provide liquidity on both sides of the LOB. It activates at a regular interval and submits n buy and sell orders some distance away from the midpoint on either side. Let **bid** and **ask** be the current bid and ask. Let m be the midpoint:

$$m = \text{floor}\left(\frac{\text{ask} + \text{bid}}{2}\right),$$

and let h be the half-spread:

$$h = \text{floor}\left(\frac{\text{ask} - \text{bid}}{2}\right).$$

The market maker submits buy orders at prices $b \in (m - h - n, \dots, m - h]$ and sell orders at prices $s \in [m + h, \dots, m + h + n)$. Each order has size 50.

Momentum agents

Similar to the market maker, the momentum agents each activate at regular intervals. When a momentum agent activates, it compares its more recent observations of the stock’s mid-price to its observations over a longer-period in order to determine the ‘momentum’ of the stock. More specifically, it compares the average mid-price m_{20} over the past 20 observations to the average mid-price m_{50} over the past 50 observations. If $m_{20} \geq m_{50}$, the agent submits a buy order at the best ask price. Otherwise it submits a sell order at the best bid price. Each momentum agent is assigned a fixed order size which is randomly drawn from the interval $[1, 10)$.

A note on wake up times Note that the momentum agents and the market maker agent each activate at regular intervals, and these intervals are the same (1-minute) in our default parameters (Table 4.1). The setup in the ABIDES platform assigns activation times at each minute for these traders and the market maker. However, the ABIDES platform also incorporate communication latency that varies across traders, which therefore affects when exactly when each agent’s orders will arrive at the exchange and based on what information. Our model does not include variable communication latency between traders and the exchange(s). Thus, in order to avoid the market maker and momentum agents being queued to act at the same nanosecond at the start of every minute, we have each of the market maker and momentum agents draw their first activation time randomly from an exponential distribution with mean equal to their wake-up interval. Even with this staggering of initial arrivals, the fixed arrival rate for the market maker and the momentum agents could still lead to recurring patterns, which we should be mindful of in our results.

4.4.2 WW+GETREAL EXPERIMENTS

The agent strategy configurations taken from Vyetenko et al. [187] are given below:

1. *base*: 5,000 noise agents, 100 value agents
2. *diverse*: 5,000 noise agents, 100 value agents, 1 market maker, 25 momentum agents

The shared parameters for these configurations are given in Table 4.1. We use these agent configurations for our alignment exercise, aiming to replicate the qualitative results from Vyetenko et al. for the same configurations in the consolidated-market setting of our model. We then examine the incremental effects of adding fragmentation, SIP latency, and latency arbitrage, as detailed in Section 4.5.

Entity	Parameter	Default	Description
Security	$\bar{r}$	10^5	Mean fundamental value
	κ	$1.67 * 10^{-12}$	Strength of mean reversion
	λ_{ms}	$2.77778 * 10^{-13}$	Megashock arrival rate
	μ_{ms}	10^3	Megashock mean
	σ_{ms}^2	$5 * 10^4$	Megashock variance
	σ_s^2	10^{-8}	OU process volatility parameter
Value	λ_v	10^{-12}	Activation rate
	$\bar{r}_v$	10^5	Value agent's view of the stock's mean fundamental value
	σ_n	10^4	Asset observation noise variance
	κ_v	$1.67 * 10^{-15}$	Value agent's view of the stock's strength of mean reversion
	σ_v^2	10^5	Estimated stock shock variance
	percent_aggr	0.1	Percent of time agent will aggress the spread
	depth_spread	2	Spread scalar for how deep to post in the book
Market Maker	num_levels	5	Number of orders to place on each side of the book
	τ_{MM}	60×10^9 (i.e., 1 minute)	How long between activations
	size	50	Order size
	last_spread	10	Spread value at initialization of strategy
Momentum	τ_{MOM}	60×10^9 (i.e., 1 minute)	How long between activations

Table 4.1: Parameters and defaults for the WW+GetReal experiments. Defaults are based on the ABIDES platform and the *random_fund_diverse* config file (https://github.com/abides-sim/abides/blob/master/config/random_fund_diverse.py).

4.4.3 PRICES, RETURNS, AND VOLUMES

The main data analyzed for each simulation run are the trades that occur and the DBBO (direct best bid and offer) series constructed by the Observer. The DBBO updates each time a new BBO update is received from an exchange. The trade prices are used to construct our price series, while the DBBO is used to calculate metrics on quote counts and quote interarrival times.

Let Δt denote a number of timesteps, indicating the level of time aggregation we are considering for a given metric. We measure price $P(t, \Delta t)$ as the price of the last trade received at or before timestep t . Thus, at any timestep without a trade, $P(t, \Delta t) = P(t - \Delta t, \Delta t)$. The first time step of the price series is taken as the first t with a trade. Returns are measured as the log-difference between subsequent prices within a price series:

$$r(t, \Delta t) = \log P(t, \Delta t) - \log P(t - \Delta t, \Delta t).$$

Volume $V(t, \Delta t)$ is measured as the number of shares traded in the period $(t - \Delta t, t]$. Note that $V(t, \Delta t) = 0$ for time steps when no trades are received; we do not forward-fill the volume from the previous step. The volume series also starts at the time of the first trade, to be consistent with the price series. Quote count $O(t, \Delta t)$ is calculated from the number of DBBO updates in a period $(t - \Delta t, t]$.

4.4.4 VERIFICATION OF WW+GETREAL INTEGRATION

We assess the fidelity of our integration of the GetReal trading strategies into our implementation of WW by aiming to reproduce the qualitative empirical-benchmarking results from Vyetrenko et al. [187] under a single-exchange configuration of our model. Vyetrenko et al. [187] do not detail their exact experimental design (e.g., how many trading days they simulated) nor exactly how they calculate the metrics. For this latter point, we can again draw from implementation details in the ABIDES platform [36]. Vyetrenko et al. compare their simulated results to empirical stock data. They specifically compare to a calendar year (2011) of data from 30 stocks for the return properties, and they use a month (June 2019) of IBM order book data from Nasdaq for the quote properties. We therefore simulate $30 \times 252 = 7560$ trading days for the return analysis and 20 days for the quote analysis.

As noted before, we also know that our market setting differs from the simulation used by Vyetrenko et al. [187] in at least some ways, which could affect the ultimate behavior measured by the empirical tests. Nonetheless, our expectation is that our implementation should produce similar qualitative results against these empirical benchmarks to the level of detail they provide in the paper. Comparing directly against their reported results provides a test of relational equivalence [13] between our model and theirs.

We begin below with the results measuring the statistical properties of the simulated returns before then examining the properties of quote dynamics. The *base* and *diverse* agent configurations in our results correspond to **config_1** and **config_2**, respectively, in Vyetrenko et al. [187]. Similar to Vyetrenko et al., all results are re-

ported visually, and we do not do any statistical tests here on the appropriateness of any distributional fits. We refer the reader to the respective figures from Vyetrenko et al. for comparison, as well as for more detail on the set of empirical benchmarks they drew from for their analyses. We discuss the empirical properties of the return series in greater detail in Section 4.5 while assessing the effects of fragmentation on these properties.

Heavy tails

Vyetrenko et al. [187] first plot histograms of their simulated return distributions alongside the distributions from their empirical data. They plot the 1-minute returns and 10-minute returns, with the expectation that the distributions exhibit heavy tails and slowly approach the normal distribution in shape as timescale increases [41, 54]. Overall, our results (Fig. 4.3) demonstrate similar qualitative behavior when comparing between the two agent configurations. Consistent with the original results, the *diverse* agent configuration has a sharper peak and less support at more extreme values relative to the *base* configuration. As timescale increases, both distributions become slightly more mound-shaped. However, we see evidence of larger returns in both distributions than shown in the original results.

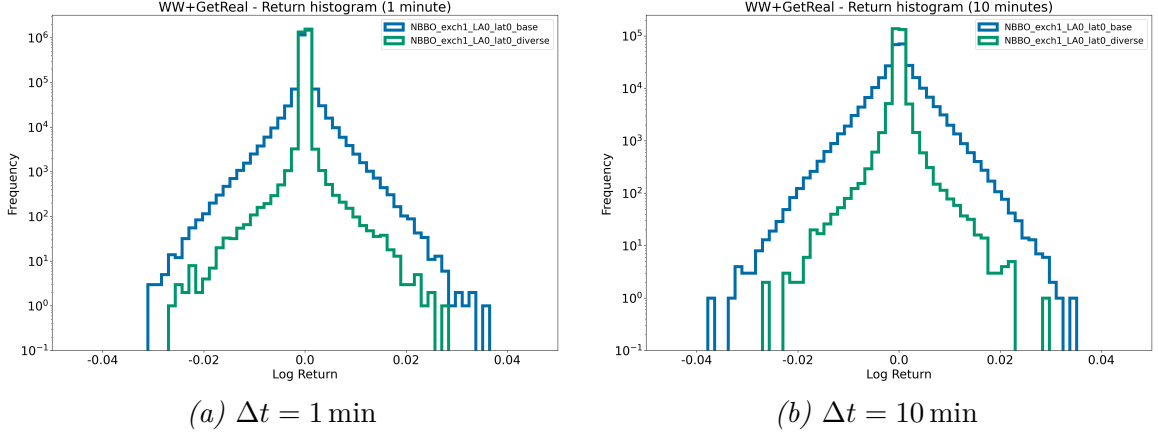


Figure 4.3: Histograms of the one-minute and ten-minute return distributions from the base and diverse agent configurations of WW+GetReal. Our results are from 7,560 runs each of the single-exchange configuration of the model. These results should be compared to the corresponding figures reported by Vyetrenko et al. [187, Figure 2, p.4].

Lack of linear autocorrelation

Vyetrenko et al. [187] next measure the linear autocorrelation of returns, which are expected to be minimal except at small timescales [54]. Vyetrenko et al. plot the distribution of first-lag local autocorrelation of the return series, calculated for 1-minute returns with a rolling window of 30-minutes. The first-lag autocorrelation results (Fig. 4.4) for the *base* agent configuration produce a mound-shaped distribution, centered around zero with tails that extend out to $\approx \pm 0.75$. This is similar to the corresponding **config_1** results from Vyetrenko et al. [187, Figure 4, p.5]. The *diverse* agent configuration meanwhile produces a more squat distribution with greater frequency of larger positive values and lower frequency of negative values than the *base* configuration. These results are again consistent with the relative relationship between configurations, but differences are evident when comparing the exact ranges shown by the Vyetrenko et al. results.

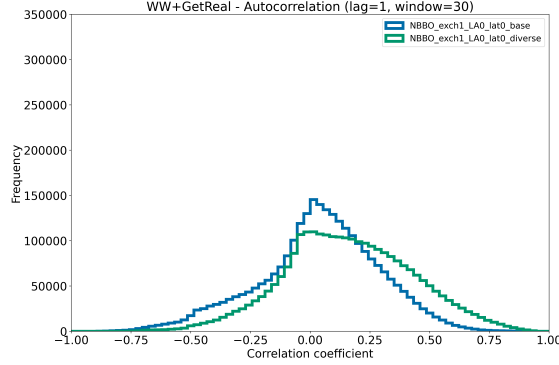


Figure 4.4: Local autocorrelation at a single lag for a rolling 30-minute window, calculated on 1-minute returns for the base and diverse agent configurations. Each set of our results are from 7,560 runs of the single-exchange configuration of the *WW+GetReal* model. These results should be compared to the corresponding figures reported by Vyetrenko et al. [187, Figure 4, p.5].

Volatility clustering

Vyetrenko et al. [187] next look at volatility clustering, which they measure by calculating the autocorrelation of squared returns:

$$C^2(\tau, \Delta t) = \text{corr}(r(t, \Delta t)^2, r(t + \tau, \Delta t)^2).$$

They calculate the squared-return autocorrelation for each return series from their simulated sample, then taking the mean of the autocorrelation results across the samples for a given agent configuration (or across their empirical samples). We find a similar qualitative relationship between the two agent configurations (Fig. 4.5), with greater squared-return autocorrelation produced by the *base* configuration for each of the 10 lags tested. Our autocorrelation values are lower in magnitude than those from the original results for both configurations, however.

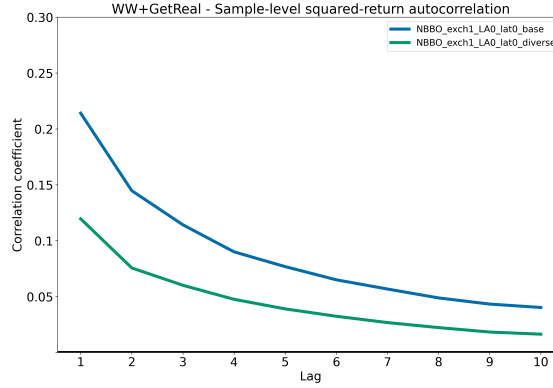


Figure 4.5: Average of the sample-level squared-return autocorrelation results for simulations of the base and diverse agent configuration of WW+GetReal. Our results are from 7,560 runs each from the single-exchange configuration of the model, with return timescale $\Delta t = 1$ min. These results should be compared to the corresponding figures reported by Vyetrenko et al. [187, Figure 4, p.5].

Volume/volatility correlation

The last empirical property of returns tested by Vyetrenko et al. [187] is volume/volatility correlation, with the expectation of a positive relation between trading volume and volatility. Similar to Vyetrenko et al. we measure volume as the total number of shares in a period. Vyetrenko et al. calculate the synchronous (lag=0) correlation between volume and volatility for each simulation as an individual sample. They then plot the histogram of the distribution of the correlation coefficient, comparing against the distribution constructed from historical stock data. We see close alignment in shape and range for both agent configurations relative to the Vyetrenko et al. results (Fig. 4.6). The frequency counts differ between our figure and theirs, although this could be due to differences in binning when constructing the histogram. Notably, many of the correlation-coefficient values are roughly zero or even negative for the *diverse* agent configuration. This is contrary to the positive relationship expected from empirical results (Section 2.3.7), but it is consistent between our results and

those from Vyetenko et al.

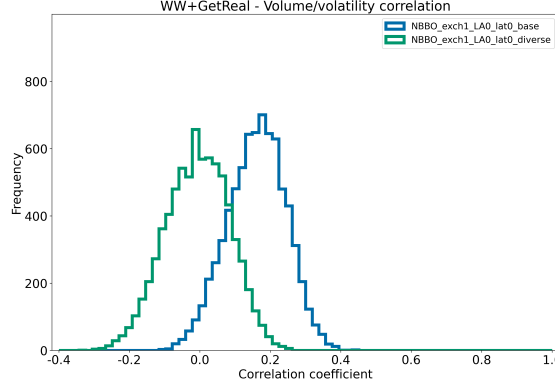


Figure 4.6: Volume/volatility correlation results for the base and diverse agent configuration of WW+GetReal. The results give the contemporaneous (zero-lag) correlation between volume and absolute returns for each independent run. Results are from 7,560 runs each of the single-exchange configuration of the model. Base timescale is $\Delta t = 1$ min. These results should be compared to the corresponding figures reported by Vyetenko et al. [187, Figure 4, p.5].

Quote counts within time window

We next move on to the dynamics of quote arrivals. Quotes in our model are captured by the DBBO produced by the Observer. Vyetenko et al. [187] report the distribution of quote arrivals in 1-minute time windows, the distribution of quote interarrival times, and the pattern of quote counts over the course of the simulated trading day.

Quote counts within 1-minute time windows are shown in Fig. 4.7 with gamma distribution fits. In our full-sample results (Fig. 4.7a), we immediately see a difference versus the results reported by Vyetenko et al. [187]: our histograms reveal outliers whereas their results do not. We identified these outliers as arising from the first activation times for the value agents, which all fall within the first 100 nanoseconds of the simulation. This first-activation logic is based on logic found in the ABIDES platform, which we detail more fully in our ODD (Appendix C.1.15).

It is not clear why similar outliers are not found in the Vyetenko et al. [187]

results. The *base* agent configuration in particular has only two agent types: noise agents and value agents. The noise agents follow a U -quadratic curve over the course of the trading day for their activation times, while the value agents follow a Poisson process after their initial activations in the first 100 nanoseconds. It would therefore make sense for the first minute of each simulation to potentially have a high amount of activity relative to the rest of the simulation. It is possible we are misinterpreting some aspect of the original code, or that there are other details affecting the distribution that prevents the first minute from being an outlier. It may also be that the volume distribution shown in the original results does not capture this outlier for some reason. It is difficult to say without rerunning their original code ourselves and repeating their analysis.

When we remove the first 100 nanoseconds of the simulation from the quote-count time series, we see much greater similarity between our results and Vyetrenko et al. [187]. These ‘trimmed’ results are shown in Fig. 4.7b. The histograms for both agent configurations have similar shapes, ranges, and frequency counts to the corresponding original results. The gamma distribution fits also follow the overall distributional shape to a similar degree as the original results. However, fully assessing the quality of the fits is difficult based only on the histograms. Overall, our quote count results closely align with the original results, but only once we remove the first 100 nanoseconds of activity.

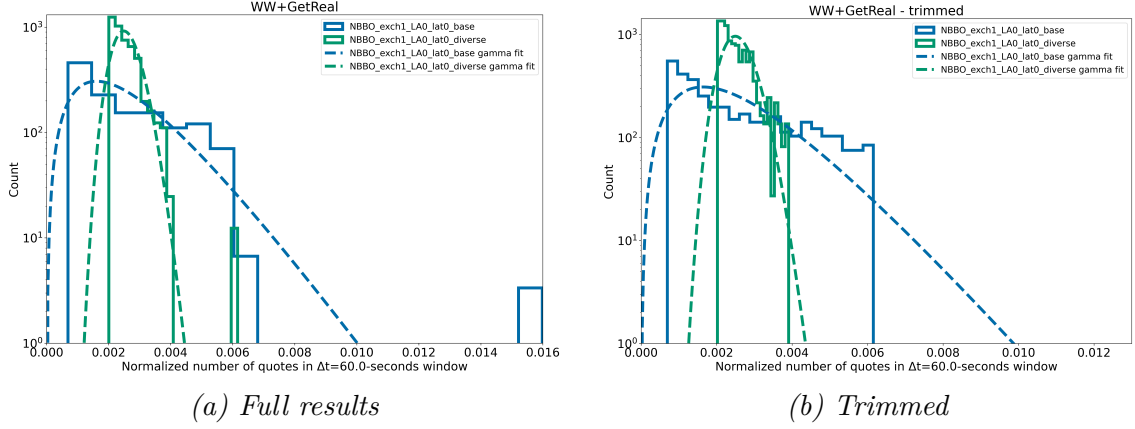


Figure 4.7: Histogram of the number of quotes in 1-minute time windows, with gamma distribution fits, for the base and diverse agent configuration of WW+GetReal. Results from 20 runs of the single-exchange configuration of the model are shown. The ‘trimmed’ results remove the first 100 timesteps (i.e., 100 nanoseconds). These results should be compared to the corresponding figures reported by Vyetrenko et al. [187, Figure 5, p.6].

Quote interarrival times

The next quote property looks at the distribution of interarrival times. Vyetrenko et al. [187] fit Weibull distributions to their experimental and historical data, finding better fits for the historical data than their simulated results. They attribute large support at zero interarrival time in their experimental results for the poor fits in their experimental results. We see a similar effect but not of the same magnitude. Following a drop after the smallest interarrival times, the density levels out around 0.2–0.3, after which it has a roughly exponential decay. This similar leveling out occurs at around 0.1 in the original results. We otherwise see similar shape and range for each of our interarrival time distributions relative to the original results.

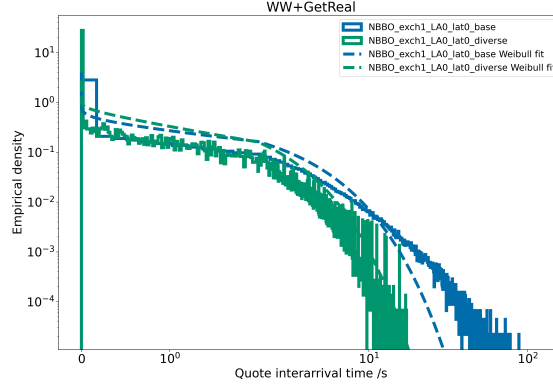


Figure 4.8: Histograms of the distribution of quote interarrival times, with Weibull distribution fits, for the base and diverse agent configurations of WW+GetReal. Our results are from 20 runs each of the single-exchange configuration of the model. These results should be compared to the corresponding figures reported by Vyetrenko et al. [187, Figure 5, p.6].

Intraday quote pattern

The last quote property reported by Vyetrenko et al. [187] tests for the presence of the U-shaped intraday activity pattern. Recall that the noise agents follow a U-quadratic distribution for their activation times, while the value agents follow a Poisson distribution with constant arrival rate. Thus, this shape is primarily driven by the activity pattern assigned to the noise agents, but it incorporates into the simulation a well-documented feature of activity found in many real-world markets [43, 197]. Vyetrenko et al. visualize this property by grouping quotes into 5-minute windows based on the time of day they were received, normalizing by the average volume level. A quadratic polynomial is then fit to the resulting data.

Similar to Vyetrenko et al. we find both distributions of intraday activity levels are well fit by quadratic polynomials, and the *diverse* agent configuration produces a flatter smile than the *base* configuration. Our results (Fig. 4.9) have similar shapes to the original results and similar relation between configurations. We see a higher

first value for the *diverse* agent configuration than seen in the original results, and we again identify this as arising from the first 100 nanoseconds of activity from the value agents.

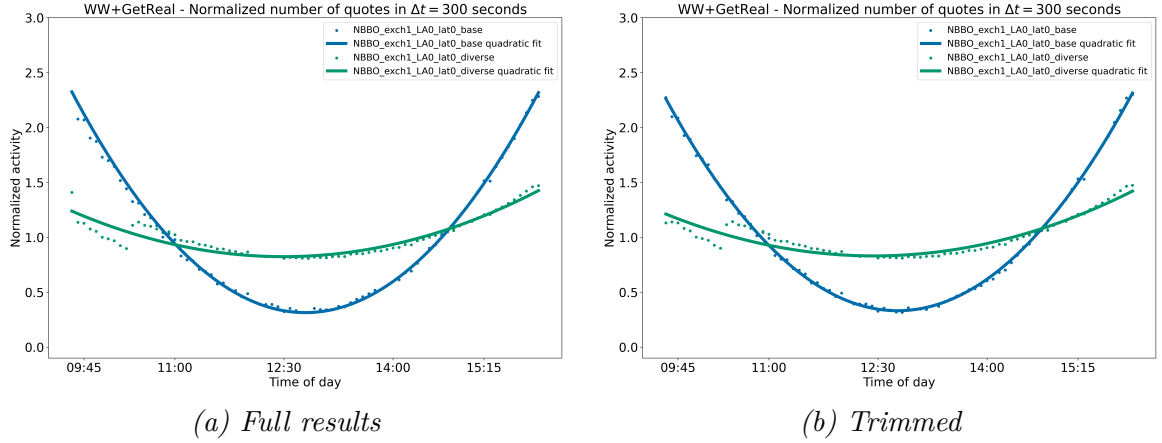


Figure 4.9: Normalized intraday quote activity level results with quadratic fits for the base and diverse agent configuration of WW+GetReal. Our results are from 20 runs each of the single-exchange configuration of the model. Time window for both sets of results is $\Delta t = 5$ min. The ‘trimmed’ results remove the first 100 nanoseconds of activity. These results should be compared to the corresponding figures reported by Vyetrenko et al. [187, Figure 5, p.6].

Verification summary

In the above results, we have tested the single-exchange configuration of our integrated WW+GetReal model against similar results from Vyetrenko et al. [187]. This provides a test of how well we have replicated the core logic that produced their main results, which were then used to assess the empirical relevance of those configurations in their simulated trading environment. We overall reproduce the qualitative relationship between the *base* and *diverse* agent configurations for each set of results. We also produce similar qualitative relationships between these simulated results and the empirical results plotted by Vyetrenko et al. We find differences in the exact ranges of values for most metrics and those shown in the original results, however, suggesting at

least some level of difference in model setup between ours and theirs. We find larger returns and lower levels of squared autocorrelation, along with a slightly different shape to the distribution of first-lag local autocorrelation values for the *diverse* agent configuration. Our quote metric results overall follow those from Vyetrenko et al., but we find sensitivity to the initial activations of the value agents in the first 100 nanoseconds of the simulation.

As noted previously, we have not adopted all logic implemented in the ABIDES [36] platform used to produce the Vyetrenko et al. results. The main known difference between our model and ABIDES is variable communication latency between entities in the simulation. The specifics of latency can impact aggregate behavior, as explored later in this study and by others [38, 66, 189]. Further analysis could be done to extend our model to include the latency settings from ABIDES and assess whether this change closes the remaining gaps in results. We discuss this more in Section 4.6.

Finally, the measures examined in this section provide insight into how well the updated model reproduces empirical behavior. A timestep in the extended model is now directly analogous to a nanosecond in clock-time, and each simulation represents a standard 6.5-hour trading day in the NMS. Having similar measures of time in the model and empirical results allows for more direct comparison between observations in the respective settings. The results shown here suggest that these agent configurations produce the qualitative behavior of volatility clustering and non-normal return distributions, although the quantitative values for the respective measures differ from the empirical referent provided by Vyetrenko et al. [187]. The quote count metrics similarly show qualitative similarity but not exact fit to the empirical results. The distributions of first-lag linear autocorrelation and volume/volatility correlation

meanwhile show poor fits relative to the empirical data. We will compare the statistical properties for the simulated returns more directly to empirical findings in later sections when testing the effects of fragmentation and latency on the simulated return distributions.

4.5 FRAGMENTATION EXPERIMENT RESULTS

We next examine how the introduction of additional exchanges and related features alter the behavior in the model. Similar to Wah and Wellman [189], we consider market configurations with either one or two exchanges. We keep all agent configuration settings consistent with our alignment exercise discussed in the previous section, except we now double the number of agents in each setting:

1. *base2x*: 10,000 noise agents, 200 value agents
2. *diverse2x*: 10,000 noise agents, 200 value agents, 2 market makers, 50 momentum agents

This choice is primarily motivated by the single market maker and odd number (25) of momentum agents in the *diverse* agent configuration. In a market setting with two exchanges, splitting the agents from the *diverse* configuration would introduce asymmetry, and in particular only one exchange would have a market maker. We find similar high-level takeaways from our experiments when using the original or the doubled (*2x*) configurations. We also find evidence of greater stability with the doubled configurations, particularly for the *diverse* agent set. We therefore focus on the results with *base2x* and *diverse2x* in our main results in Section 4.5 and provide results for *base* and *diverse* in our supplemental material.

In addition to the number of exchanges in the market, we also vary the amount of SIP latency. We base our latency settings on reported detail from the NMS. As mentioned earlier, the SIPs in the NMS report average processing latency of around 10–20 μ s for the first quarter of 2026 [59, 182]. Similar to the type of latency represented in the model, these empirical measures of latency only capture the amount of time that elapses between when the SIP receives a quote and when the SIP has published its update to the NBBO. They do not include geographic latency nor the amount of time it takes an exchange to publish the quote to the SIP in the first place. We consider SIP latency amounts $\delta \in \{0, 10, 100, 1000\} \mu$ s for our experiments. Finally, we consider the incremental effect of adding the LA agent in settings with multiple exchanges and nonzero latency.

We conduct experiments for each agent configuration, varying the number of exchanges $N_{exch} \in \{1, 2\}$, the amount of SIP latency δ , and the presence of the LA agent $N_{LA} \in \{0, 1\}$. We simulate 1,000 runs for each agent configuration and model setting. Random seeds are controlled in the model such that agent activations and the exogenous fundamental value are consistent for different market settings that have the same agent strategy configuration. This allows us to directly examine, for a given seed, the incremental effects on the price path of introducing an additional exchange, latency, and/or the LA agent. We provide examples of these effects below for individual random seeds before providing aggregate statistics across all runs and settings.

We visualize the price paths and returns over time, as well as the deviation of prices away from the price of the consolidated-market baseline. We show results for $N_{exch} \in \{1, 2\}$ and $\delta \in \{0, 10 \mu$ s $\}$. In the results under the BBO information regime,

we do not show the consolidated market with latency, as latency has no effect in that setting. We do not include results with the LA agent in these example figures due to space. The relative absolute deviation (RAD) [168] is measured as:

$$RAD(t, \Delta t) = \frac{|P^*(t, \Delta t) - P^C(t, \Delta t)|}{P^C(t, \Delta t)},$$

where $P^C(t, \Delta t)$ is the price of the consolidated-market setting without latency at time t and $P^*(t, \Delta t)$ is the price of some other market setting at the same time. Timescale is $\Delta t = 1$ min for the prices and returns plotted here.

Both information regimes produce clear evidence that the price paths are not identical across market configurations. The distinction is much clearer for the BBO information regime, where we see larger swings and extended periods of volatility. These plots also highlight nuances of the simulation settings and possible evidence of unrealistic behavior. Some large price changes for the *base2x* agent configurations under the BBO information regime exhibit large rises followed by similarly large drops, or vice versa. The *diverse2x* agent configuration produces very constrained price changes in the consolidated market or each configuration under the NBBO information regime. When the *diverse2x* agents instead look only at their local BBOs in a fragmented market, the price varies more widely, with swings similar to the *base2x* prices under this information setting. The *base2x* agent configuration under the BBO information regime produces much larger returns in a fragmented market setting than is seen for any of the other settings. These nuances should be kept in mind as we delve into our later results.

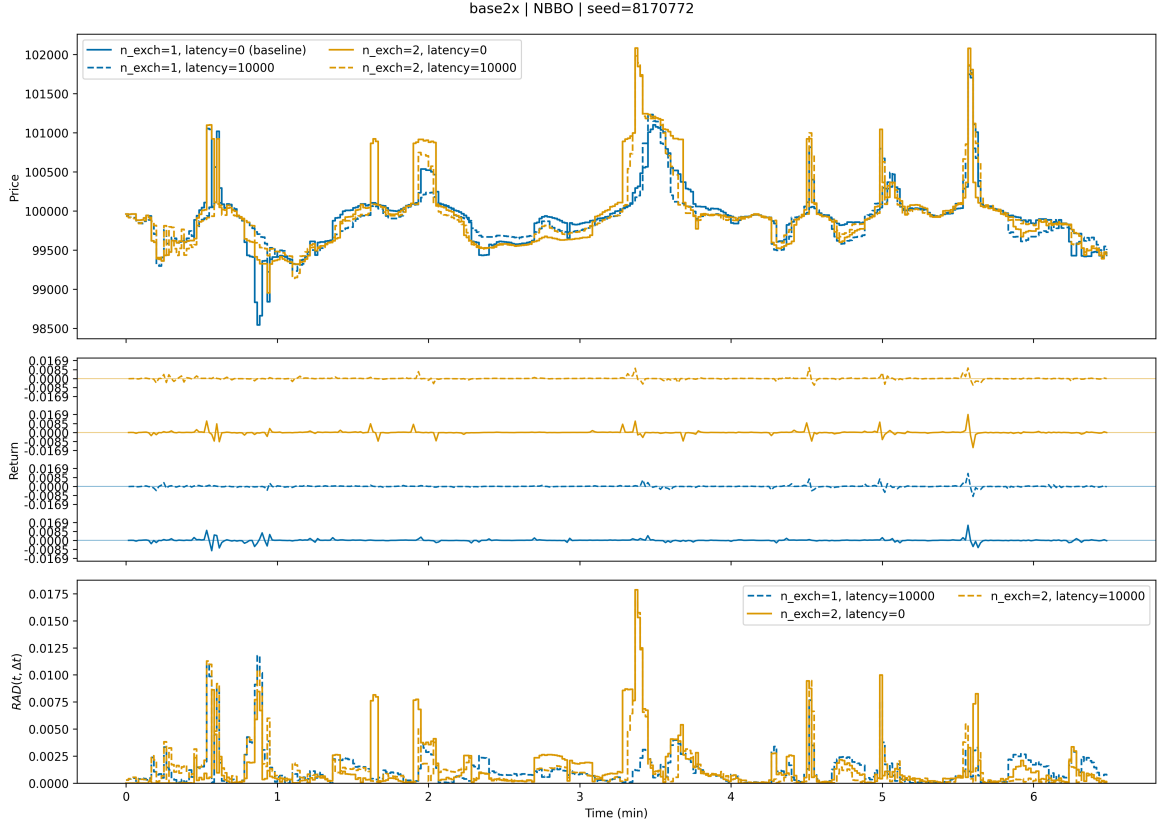


Figure 4.10: Paired price paths for the base2x agent configuration under the NBBO information regime with a common random seed (8170772). Prices are simulated under consolidated and fragmented ($N_{exch} = 2$) markets for zero and nonzero latency ($\delta = 10 \mu s$). The first plot charts the 1-minute prices over time for each setting. The second plot shows the corresponding 1-minute returns. The third plot shows the relative absolute deviation of each price versus the price of the consolidated-market setting without latency. We do not plot the deviation of the consolidated-market setting against itself, as it would always be zero.

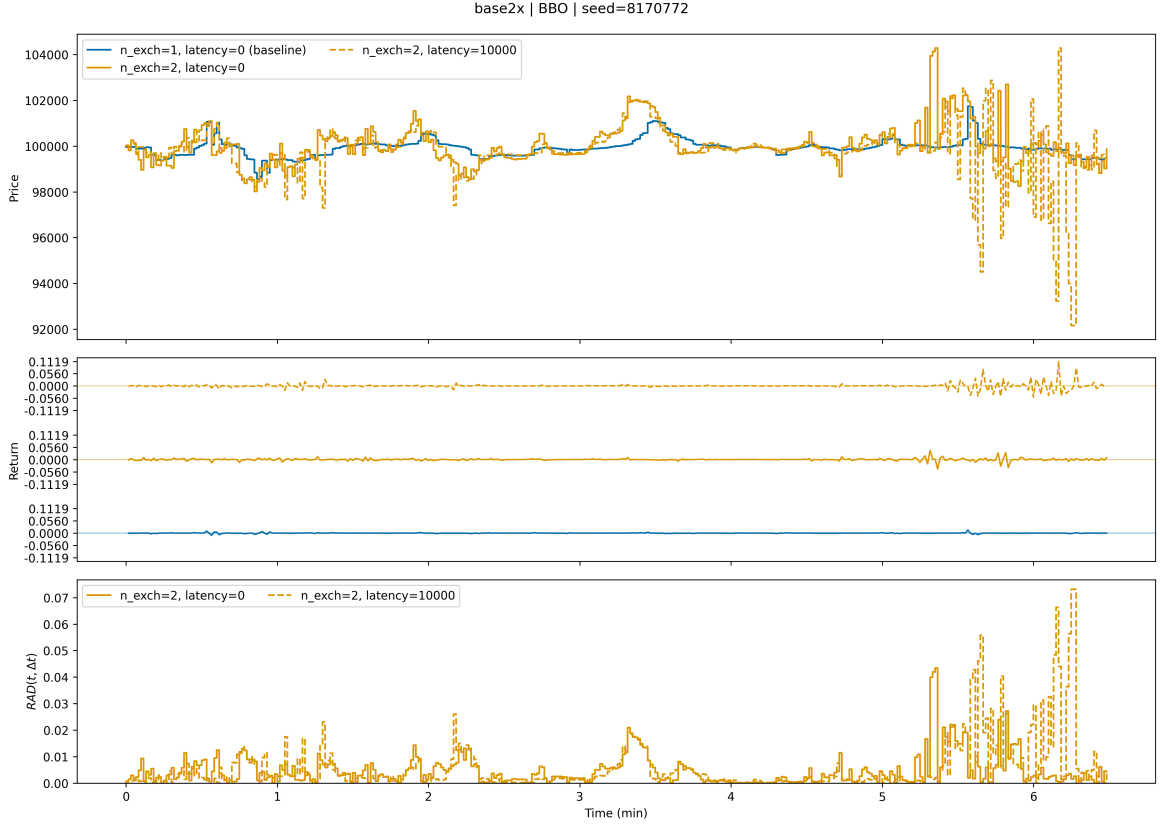


Figure 4.11: Paired price paths for the *base2x* agent configuration under the BBO information regime with a common random seed (8170772). Prices are simulated under consolidated market setting without latency and fragmented ($N_{\text{exch}} = 2$) market settings with zero and nonzero latency ($\delta = 10 \mu\text{s}$). Nonzero latency has no effect for the consolidated market in the BBO information regime, so we do not include it here. The first plot charts the 1-minute prices over time for each setting. The second plot shows the corresponding 1-minute returns. The third plot shows the relative absolute deviation of each price versus the price of the consolidated-market setting without latency. We do not plot the deviation of the consolidated-market setting against itself, as it would always be zero.

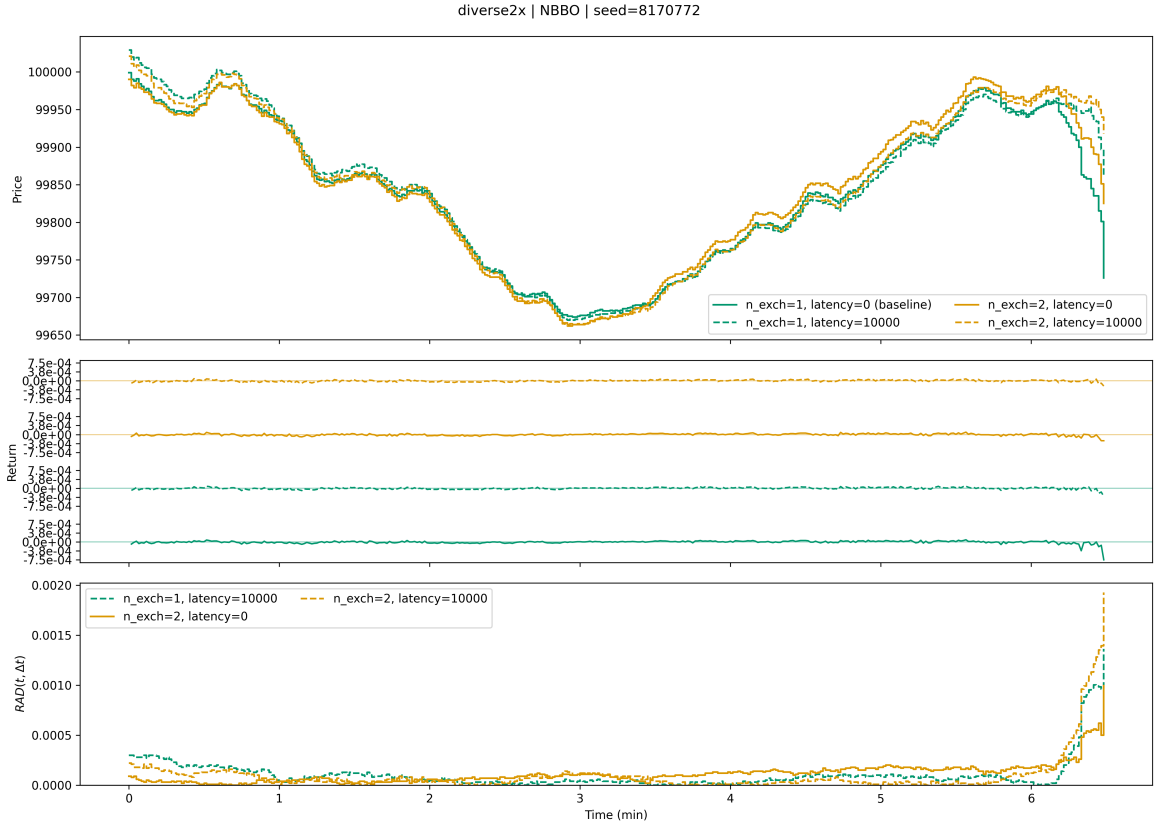


Figure 4.12: Paired price paths for the diverse2x agent configuration under the NBBO information regime with a common random seed (8170772). Prices are simulated under consolidated and fragmented ($N_{exch} = 2$) markets for zero and nonzero latency ($\delta = 10 \mu s$). The first plot charts the 1-minute prices over time for each setting. The second plot shows the corresponding 1-minute returns. The third plot shows the relative absolute deviation of each price versus the price of the consolidated-market setting without latency. We do not plot the deviation of the consolidated-market setting against itself, as it would always be zero.

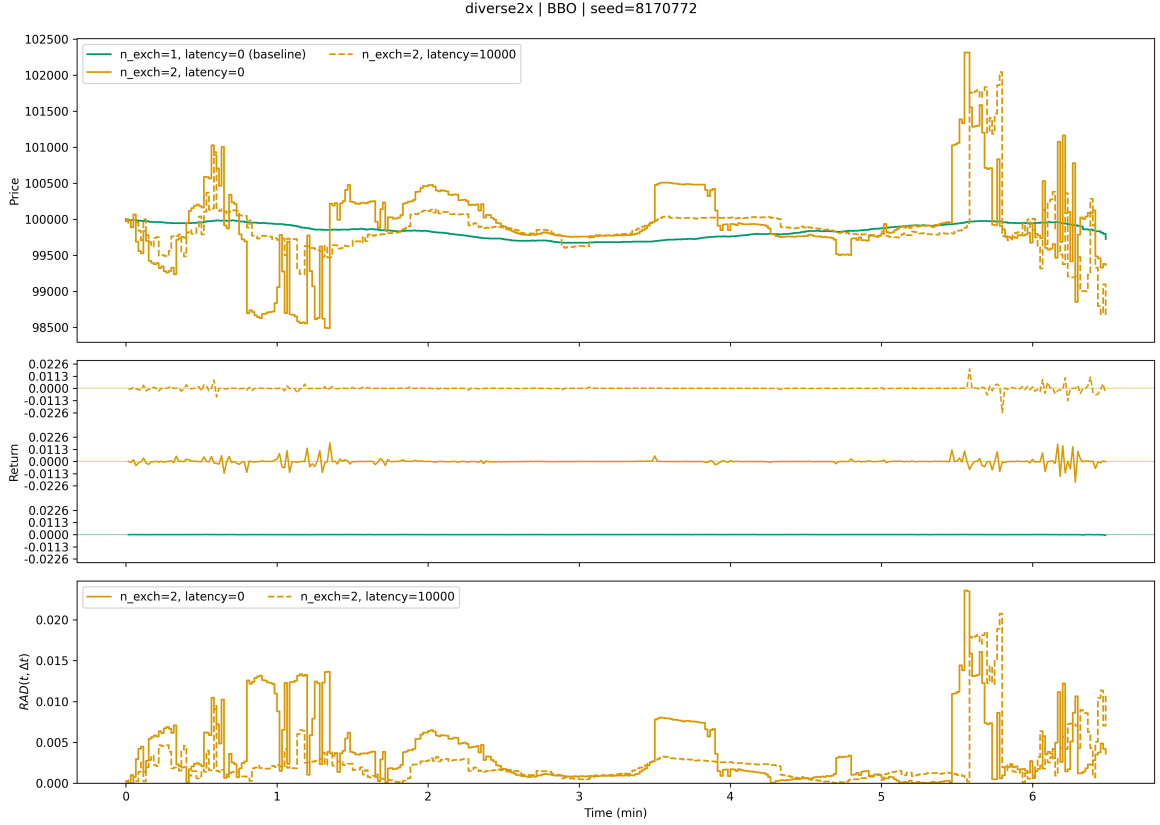


Figure 4.13: Paired price paths for the *diverse2x* agent configuration under the BBO information regime with a common random seed (8170772). Prices are simulated under the consolidated market setting without latency and fragmented ($N_{\text{exch}} = 2$) market settings with zero and nonzero latency ($\delta = 10 \mu\text{s}$). Nonzero latency has no effect for the consolidated market in the BBO information regime, so we do not include it here. The first plot charts the 1-minute prices over time for each setting. The second plot shows the corresponding 1-minute returns. The third plot shows the relative absolute deviation of each price versus the price of the consolidated-market setting without latency. We do not plot the deviation of the consolidated-market setting against itself, as it would always be zero.

These examples demonstrate that varying the structure of the modeled market can produce different price series while keeping all agent settings consistent and controlling for randomness. We next aim to quantify by how much the structural features considered cause prices to diverge relative to the baseline setting of a consolidated market with no latency.

4.5.1 PRICE DEVIATION FROM CONSOLIDATED BASE-LINE

For a given agent configuration and information regime, let $P_i^C(t, \Delta t)$ denote the price at time t , measured at timescale Δt , for the consolidated-market setting with no latency, simulated under random seed i . Let $P_i^*(t, \Delta t)$ be the corresponding price for some other model setting with the same agent configuration, information regime, and random seed i . Then the RAD between P_i^* and P_i^C over the course of the simulation is:

$$RAD_i^{*,C}(\Delta t) = \frac{1}{N_i^{*,C}} \sum_t \frac{|P_i^*(t, \Delta t) - P_i^C(t, \Delta t)|}{P_i^C(t, \Delta t)},$$

where $N_i^{*,C}$ is the number of common periods for the two simulations being compared. Note that the first price in each series is for the first period with a trade. After that first period, recall that we forward-fill price for any periods lacking trades up through the last period with a trade for the simulation. We therefore limit both series to the periods $t_{\min} \leq t \leq t_{\max}$, where $t_{\min}$ is the first period once both simulations have had at least one trade and $t_{\max}$ is the period where the first simulation to end had its last trade. For the common periods where we have a measure of price, $RAD_i^{*,C}(\Delta t)$ measures the average absolute-percentage deviation of the price process under setting $*$ from the price under the consolidated benchmark. We calculate the average $RAD^{*,C}$ across the R random seeds for which we have simulations:

$$RAD^{*,C}(\Delta t) = \frac{1}{R} \sum_{i=1}^R RAD_i^{*,C}(\Delta t).$$

We then calculate 95% bootstrap confidence intervals for the metric. A bootstrap sample is constructed by sampling, with replacement, R random seeds, recalculating $RAD^{*,C}$ for those seeds—call that value $RAD_b^{*,C}$ for bootstrap b —and then calculating the confidence interval from the set $\{RAD_b^{*,C}\}_{b \leq B}$.

Our RAD calculation is based on the metric introduced by Stöckl, Huber, and Kirchler [168] to measure mispricing of an asset relative to its fundamental value in an experimental market. Stöckl et al. justify their formulation of RAD based on the following qualities: it (i) relates price to the fundamental value, (ii) is monotone in their difference, and (iii) is independent of the number of periods and the level of the fundamental value. Similarly, our $RAD^{*,C}$ measure relates the price from one configuration of the model to the price from another (the consolidated-market baseline), is monotone in their difference, and is independent of the number of periods and the average baseline price. This allows us to compare mean RAD across different structural settings as a measure of how much these settings alter the price series relative to the consolidated baseline. We do not claim that the consolidated-baseline price is ‘correct’ in any theoretical sense. Our analysis here is rather intended to measure the extent simulated price series differ under different structural settings.

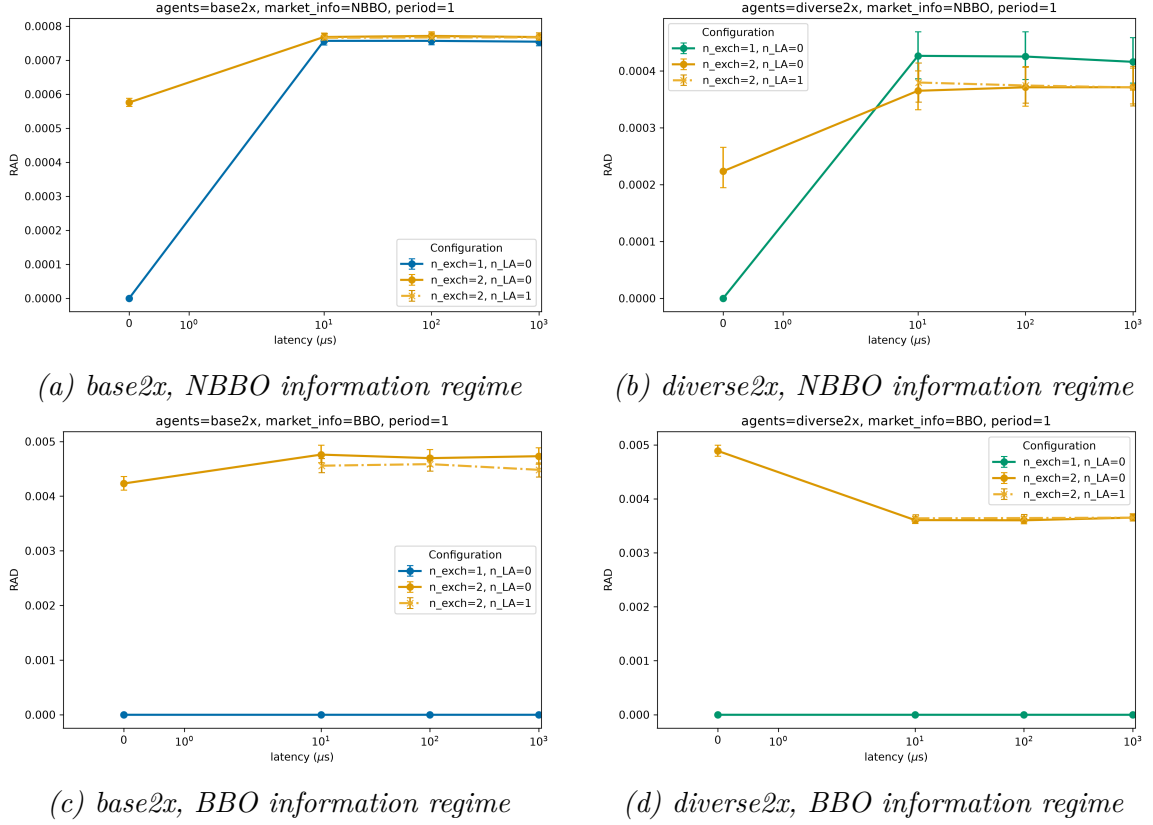


Figure 4.14: Relative absolute deviation (RAD) versus consolidated-market baseline without latency.

Prices diverge most from the consolidated baseline when traders have a fragmented view of market information. RAD from fragmented markets under the BBO information regime is roughly an order of magnitude greater than the differences found under other settings in the NBBO information regime. SIP latency has mixed effects under the BBO-based information setting, where it only affects the exchanges' rerouting decisions. Under this information setting, nonzero latency causes RAD to increase for the *base2x* results and decrease for the *diverse2x* results relative to the respective fragmented-market results without latency. Note that SIP latency has no effect in the consolidated market setting when agents use the exchange BBO for their information,

and thus RAD is zero under each latency value for that information setting. Under the NBBO information regime, nonzero SIP latency produces greater RAD than venue fragmentation alone. Venue fragmentation with no change to what quotes traders have access to produces the least deviation from the consolidated baseline. Finally, note that we find minimal effect from the presence of latency arbitrage—the RAD for the fragmented market settings with nonzero latency is similar with or without the LA agent in each setting.

Together, these results provide a hierarchy of which structural features in the model cause prices to diverge most from the consolidated-market setting with no latency. Changes to the information traders act based on have the greatest effects, with fragmented market information producing much larger effects than any other setting overall. Delayed access to the NBBO has a larger effect than venue fragmentation alone. Venue fragmentation without any change to the price quotes traders see causes prices to diverge the least. And latency arbitrage has minimal incremental effects relative to the fragmented market setting without the LA agent.

4.5.2 FRAGMENTATION AND THE STATISTICAL PROPERTIES OF RETURNS

With the RAD results as context, we now examine how the identified price divergences correspond with aggregate differences in the statistical properties of returns generated under different structural contexts. Given the similarity in RAD for different levels of nonzero latency δ , we focus on results here with $\delta \in \{0, 10\mu s\}$. We compare properties of the return distributions generated for the following settings under each

of the agent configurations and information settings:

1. Consolidated market with no latency
2. Consolidated market with latency ($\delta = 10 \mu s$)
3. Fragmented market ($N_{\text{exch}} = 2$) with no latency
4. Fragmented market ($N_{\text{exch}} = 2$) with latency ($\delta = 10 \mu s$)
5. Fragmented market ($N_{\text{exch}} = 2$) with latency ($\delta = 10 \mu s$) and LA agent

We plot the absolute return distributions and calculate metrics commonly used to test for evidence of empirical properties of returns—heavy tails, aggregational Gaussianity, lack of linear autocorrelation, volatility clustering, and volume/volatility correlation.

We find further evidence that changes to the information acted on by traders affects aggregate price dynamics more than just the fragmentation of orders across venues. When traders are limited to only information from their assigned exchange, the introduction of an additional exchange alters the qualitative behavior of the return distribution, producing heavier tails, increased volatility clustering, and less persistent linear autocorrelation. Nonzero NBBO latency alters the level of kurtosis and correlation metrics in some cases, while the introduction of the LA agent has minimal additional effects. Differences are also found from the introduction of an additional exchange under the NBBO information regime, but most of these effects are smaller than the introduction of latency under this consolidated-information setting.

Distributional differences

We plot the complementary cumulative distribution functions (CCDF) of the absolute return distributions at 1-minute and 60-minute timescales—Figures 4.15 and 4.16 for

the NBBO and BBO information regimes, respectively. Gaussian distributional fits are provided for the return distribution from the consolidated market without latency. The *base2x* and *diverse2x* agent configurations produce different distributional shapes under all settings. As shown by Vyetrenko et al. [187] and our alignment results (Section 4.4.1), the *diverse2x* return distributions are sharply peaked, with much of the density concentrated at very small magnitudes. Vyetrenko et al. speculate that the market maker may overly reduce volatility, thus producing a less realistic distribution.

Introducing NBBO latency and/or additional exchanges does not materially change the shape of the distributions under the NBBO information regime. Fragmentation under the BBO information regime, however, greatly increases the tails of the distribution and alters the shape overall. The *base2x* agent configuration produces very large 1-minute returns under this information setting, the magnitude of which may indicate instability. An absolute return of ≈ 10 from the starting price of 100000 means the asset would have lost almost all its value or increased by orders of magnitude. The restriction on volatility from the market maker may help the stability of the *diverse2x* agent configuration under the BBO information regime. The largest 1-minute returns are approximately 0.8–0.9 for the fragmented market without latency and 0.08–0.1 when latency is introduced.

One possibility for fragmented markets under the BBO information regime is that the two exchanges operate almost independently, with only loose coupling of dynamics due to exchanges occasionally rerouting orders to prevent trade-throughs. Note, however, that each exchange under the *base2x* configuration has the same number and type of agents as the consolidated exchange under the original *base* configuration. We

ran the *base* configuration for more than 1,000 runs in a consolidated-market setting in Section 4.4.1. We did not see returns of nearly the magnitude seen for the *base2x* under that prior analysis, suggesting that the returns observed here are not simply due to the price going up at one exchange while it goes down at the other. This may indicate that the coupling between the markets produces feedback dynamics that are different from what the agents would produce at each exchange if no orders were rerouted. Future research could test this more directly.

The distributions approach the normal distribution as timescale increases but still exhibit tail behavior beyond the normal distribution. This property is known as ‘aggregational Gaussianity’ [54]. The *base2x* return distribution from the consolidated market settings or under the NBBO information regime are closest to normal at this timescale. The fragmented-market returns under the BBO information regime approach normality more slowly, and extreme values from the *base2x* agent configuration are still seen at the 60-minute timescale when the LA agent is present.

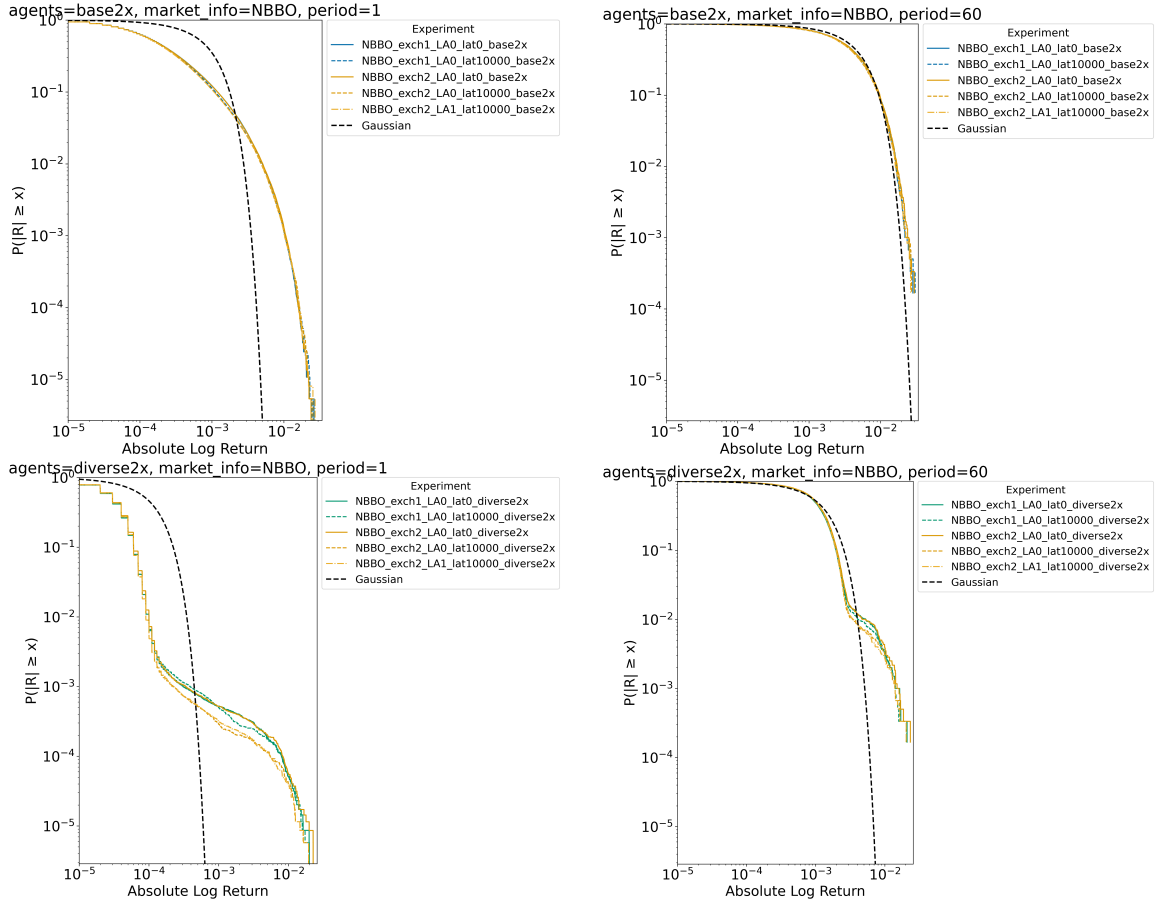


Figure 4.15: CCDFs of the absolute return distributions generated by market configurations where the trader agents use the SIP NBBO for their market information. Results are from 1000 runs of each experiment. Latency values included in these results are 0 and 10 μ s, written in nanoseconds in the experiment key.

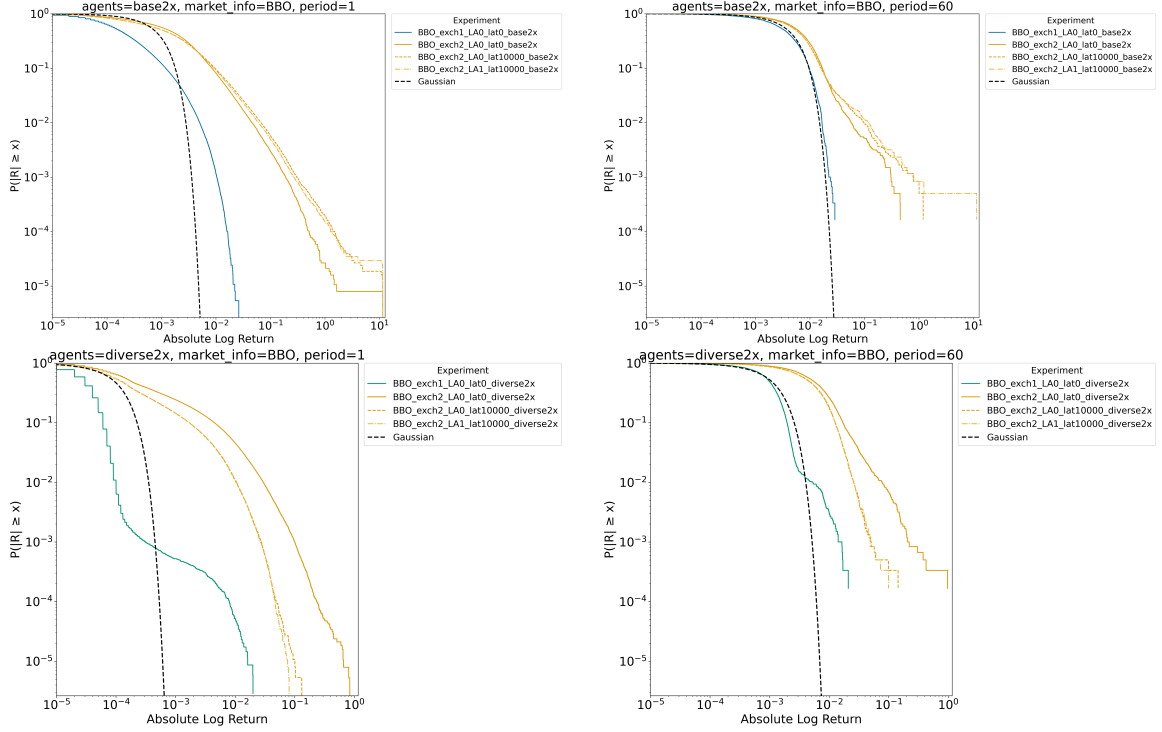


Figure 4.16: CCDFs of the absolute return distributions generated by market configurations where the trader agents use their local exchange BBO for market information. CCDFs are given for the 1-minute (left) and 60-minute (right) absolute return distributions. Results are from 1000 runs of each experiment. Latency values included in these results are 0 and 10 μ s, written in nanoseconds in the experiment key.

As a complement to the distributional figures just discussed, we plot kurtosis as a function of timescale in Fig. 4.17. As discussed in Chapter 2, kurtosis is frequently used as a statistical measure of heavy tails. Positive kurtosis indicates a distribution displays a sharper peak and heavier tails than a normal distribution. However, because it also measures the peakedness of the distribution and not just the tails, kurtosis does not provide a pure measure of tailedness. Comparing kurtosis values between two non-normal distributions also does not provide a clear indication of which has heavier tails. With these caveats, kurtosis is still a useful measure of the shape of a distribution, and comparing kurtosis across model configurations or timescales

can show changes in this shape.

Kurtosis decreases as timescale increases for most settings (Fig. 4.17), consistent with the property of aggregational Gaussianity (see Section 2.3.2). The exception is the *diverse2x* fragmented-market setting without latency under the BBO information regime, for which we see kurtosis increase as timescale goes from 1-minute to 10-minutes before coming back down. As with our previous results, kurtosis is greatly affected by fragmentation under the BBO information regime and more modestly affected under the NBBO information regime. Interestingly, kurtosis increases with fragmentation under the BBO information regime for the *base2x* agent configuration, but it decreases with fragmentation under the *diverse2x* agent configuration in most settings and timescales measured. From the CCDFs above, we see that the high kurtosis for the consolidated-market setting under the *diverse2x* agent configuration is likely due to the peakedness of the return distribution. This provides a demonstration of why kurtosis is not a pure measure of tail heaviness. Fragmentation under the BBO information regime also affects the rate at which kurtosis decreases as a function of timescale, with the kurtosis from the fragmented-market results decreasing more slowly than the consolidated-market return distributions. Latency and latency arbitrage have mixed effects, increasing kurtosis in some settings and timescales while decreasing it in others.

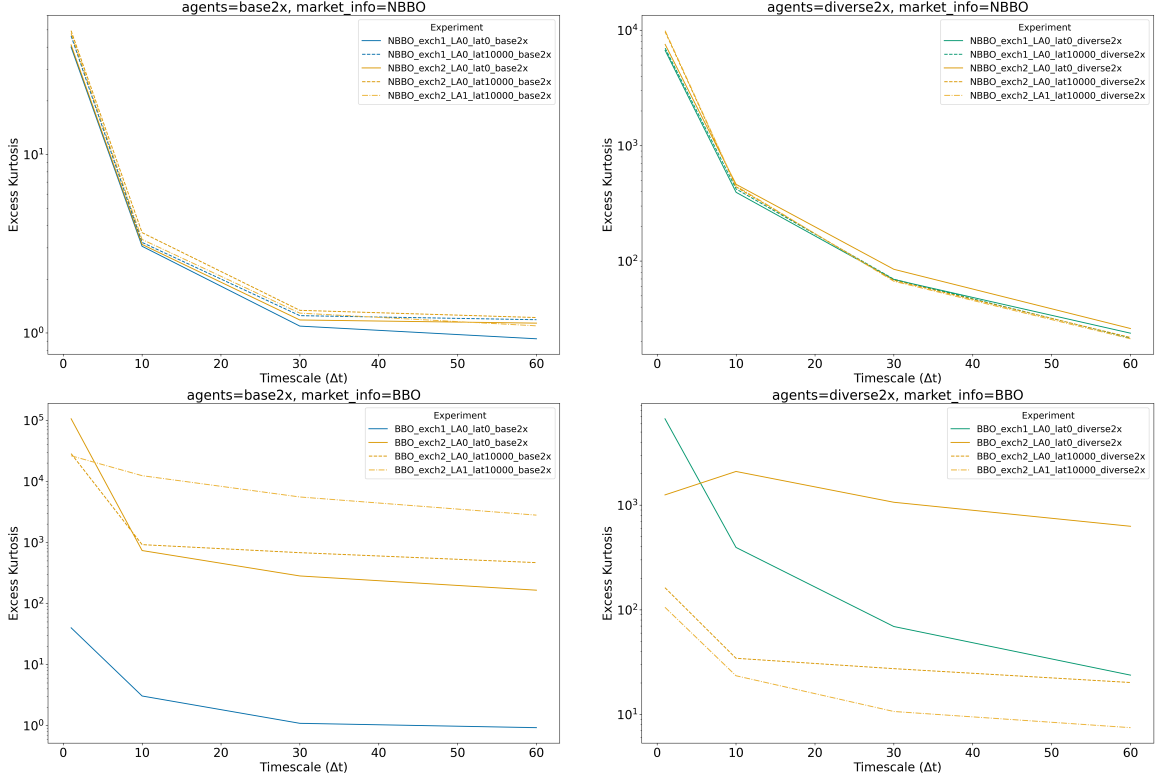


Figure 4.17: Kurtosis as a function of timescale (in minutes). Results are shown for the base2x and diverse2x agent configurations under the NBBO and BBO information regimes. Results are from 1000 runs of each experiment. Latency values included in these results are 0 and 10 μ s, written in nanoseconds in the experiment key.

Linear autocorrelation

Next, we measure the linear autocorrelation of returns under the different market settings. As mentioned earlier during the Vyetrenko et al. [187] benchmarking exercise, returns are expected to exhibit minimal linear autocorrelation except at small intraday timescales [54]. The linear autocorrelation function (ACF) is:

$$C(\tau, \Delta t) = \text{corr}(r(t, \Delta t), r(t + \tau, \Delta t)),$$

taking the Pearson correlation of the return series against a lagged version of itself. Cont [54] in 2001 listed $\simeq 20$ minutes as the timescale below which ACF results may be non-trivial due to microstructure effects. More recently, in Chapter 2 we found autocorrelation of 1-minute returns for Dow 30 stocks circa 2018–2019 was less than -0.1 at the first lag and fluctuated around zero for subsequent lags.

In our analysis in Chapter 2, we normalized the empirical return series to account for intraday seasonality before calculating the linear ACF and most other results. Our return series exhibit a similar ‘u-shaped’ intraday pattern of activity, as demonstrated earlier when reproducing the Vyetenko et al. [187] results. We therefore normalize our return series similar to the methodology detailed in Section 2.2, as not doing so may affect the stability of our measure and how directly we can compare to empirical results. We normalize the returns by first centering and standardizing each simulation’s returns by the mean and standard deviation at the given timescale. We then divide each return by the average absolute return at that time of day across the simulations for the given experimental setting.

In the baseline (consolidated-exchange) results and the fragmented results under the NBBO information regime, we find patterns of persistent autocorrelation. This pattern is relatively weak for the *base2x* agent configuration. Linear ACF from the *diverse2x* agent configuration stays above 0.1 for more than 40 lags and positive for more than 60 lags, however. This is in contrast to empirical results of linear ACF in the NMS (Figure 2.2), and it is also higher than the corresponding level of absolute autocorrelation found later on in this section. The earlier results from Vyetenko et al. [187] (and our replication, Fig. 4.4) show the distribution of first-lag local autocorrelation was more positively skewed than similar results for historical stock

data. The unrealistic pattern of persistently positive ACF was not apparent from this earlier metric, however. We find that the BBO information regime dampens these effects and brings the level of autocorrelation more in line with empirical results. The first-lag ACF results under the BBO information regime are also negative at the first lag, as is frequently found at intraday timescales.

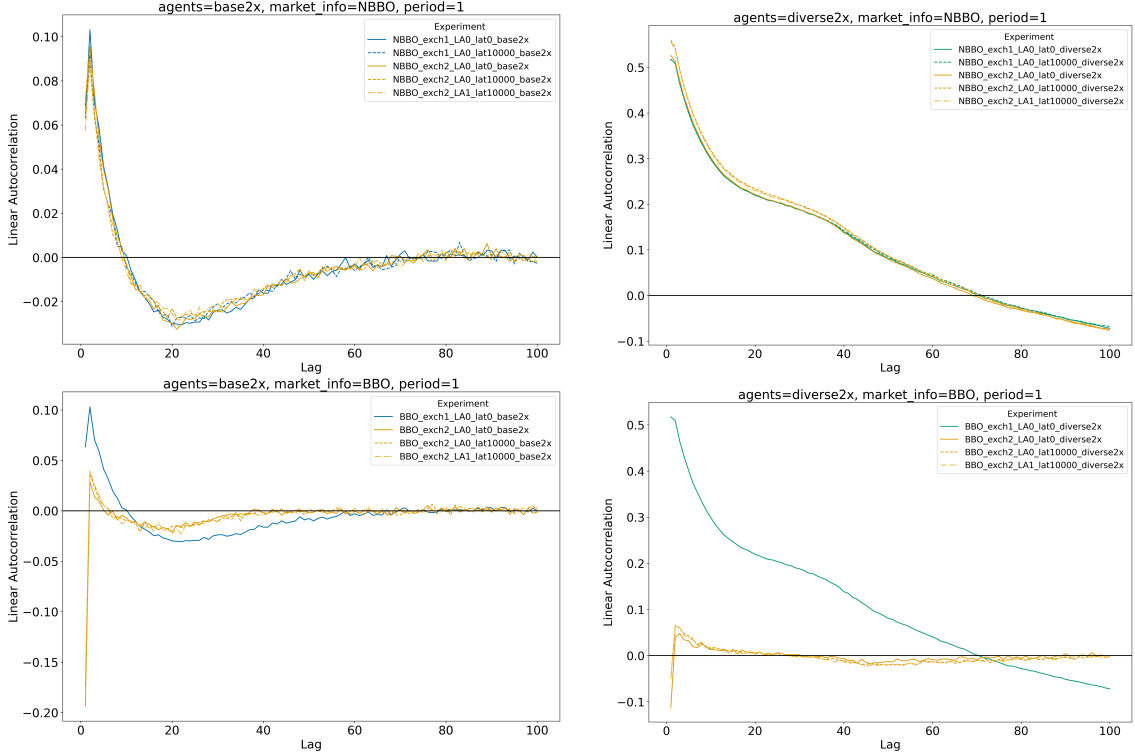


Figure 4.18: Linear autocorrelation of the 1-minute returns for the base2x and diverse2x agent configurations with different market information settings. Results are from 1000 runs of each experiment. Latency values included in these results are 0 and 10 μ s, written in nanoseconds in the experiment key.

Volatility clustering

In contrast to the lack of linear autocorrelation, the magnitudes of returns are expected to be positively autocorrelated, with a slow decay as lag increases [54]. The effect is more specifically expected to exhibit an asymptotic decay, roughly fit by a

power law (as discussed in Chapter 2). We measure these properties here by calculating the autocorrelation of the absolute return series. We again use the normalized return series for this analysis to account for intraday seasonality. We calculate the absolute autocorrelation for up to 100 lags to assess how the effect decays as a function of lag.

We overall find evidence of volatility clustering and slow-decay of absolute autocorrelation for each set of results (Fig. 4.19). The level of absolute autocorrelation found and the shape of its decay as a function of lag are influenced by the structural settings, particularly when comparing the consolidated- and fragmented-market results under the BBO information regime. Fragmentation under this information setting greatly increases the absolute autocorrelation and produces a slower decrease as a function of lag. The fragmented markets without latency for both agent configurations produce absolute autocorrelation values that start and remain much higher than empirical levels found for similar metrics (Figure 2.7).

Latency further increases the absolute autocorrelation for the *base2x* agent configuration but dampens it for the *diverse2x* agent configuration. Rather than the expected asymptotic decay of the effect, however, the absolute autocorrelation decreases steadily and even goes negative eventually for each configuration. A true power-law decay is not possible when measuring intraday returns at a finite timescale—there are only 390 minutes in the trading day—but our empirical results in Section 2.3.4 more closely follow the expected power-law decay for ≈ 100 lags or more. See our supplemental results in Figure A.19 for analogous empirical results in linear scale (versus the log-log scale plots shown in Section 2.3.4).

Under the NBBO information regime, the market configurations with latency

and/or fragmentation stay close to the consolidated-market baseline. The level of absolute autocorrelation for the *diverse2x* results goes to approximately zero after roughly 30 lags, after which point it exhibits an interesting shape before eventually going negative. Under this consolidated information setting, little overall effect from fragmentation and/or latency is found for the overall shape of decay and level of absolute autocorrelation for either agent configuration.

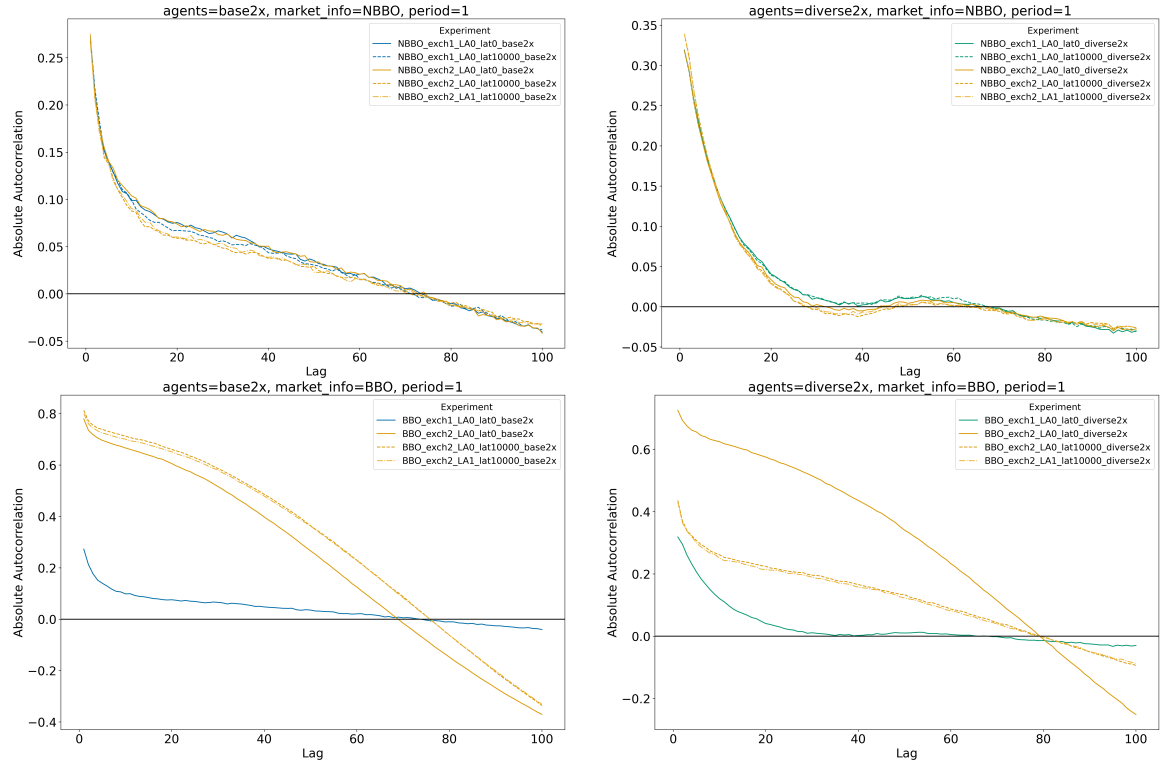


Figure 4.19: Autocorrelation of the absolute 1-minute returns for the *base2x* and *diverse2x* agent configurations with different market information settings. Results are from 1000 runs of each experiment. Latency values included in these results are 0 and 10 μ s, written in nanoseconds in the experiment key.

Volume/volatility correlation

Finally, we calculate the correlation between volume and volatility, which are expected to be positively related, at least synchronously (see Section 2.3.7). We measure this property here by calculating the correlation for all samples under a given experimental setting, versus the sample-level correlation calculated for our alignment exercise against Vyetenko et al. [187] (Section 4.4.1). We normalize the volume series by the intraday activity pattern, analogous to how we normalize the returns and volatility series. We center and standardize the volume from each simulation before dividing the volume at a given time by the average amount of volume at that time across the simulations for that experiment.

Overall, the volume/volatility correlation is found to be relatively weak compared to our empirical results in Figure 2.9. In many of the experimental results, the synchronous (zero-lag) correlation is negative, in contrast with the expected relationship. Fragmentation alters the strength of the correlation under the BBO information regime, but this effect does not necessarily bring the correlation more in line with empirical results. In the *diverse2x* agent configuration results, the synchronous correlation between volume and volatility is positive for each setting but weaker in the fragmented-market results than for the consolidated-market setting. Fragmentation meanwhile changes the relationship from negative to weakly positive for the *base2x* agent configuration under the BBO information setting. Under the NBBO information setting, fragmentation and latency have little effect for the *base2x* agent configuration, whereas nonzero latency increases the volume/volatility correlation for the *diverse2x* agent configuration.

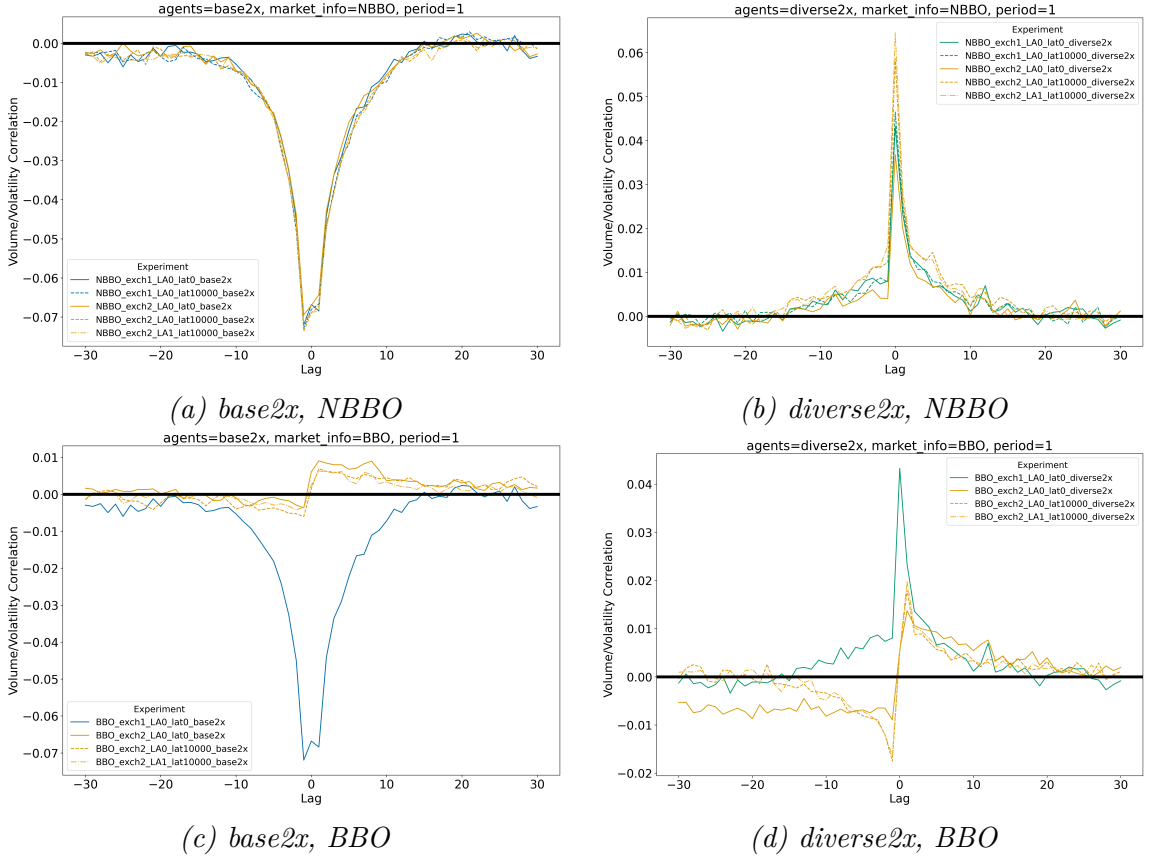


Figure 4.20: Volume/volatility correlation results for the base2x and diverse2x agent configurations with different market information settings. Results are from 1000 runs of each experiment. Latency values included in these results are 0 and 10 μ s, written in nanoseconds in the experiment key.

4.6 DISCUSSION

We demonstrate in this study that fragmented market structure is sufficient in our model setting to generate changes in aggregate price dynamics and that these effects are most acute when market information is itself fragmented. Under settings where agents act based only on the price quotes from their local exchange, fragmentation materially affects the qualitative properties of returns, producing heavier tails, stronger volatility clustering, and less persistent linear autocorrelation. Incremental

effects of SIP latency and latency arbitrage are found, but these effects are not uniform across agent configurations or information settings. Of the structural changes tested here, fragmentation of trading without altering the information environment has the smallest effects relative to the consolidated-market without latency. The addition of the LA agent to fragmented market settings with latency also has minimal incremental effects for most metrics.

As noted previously, the information settings we examine here represent two extremes for how traders could access price quotes in a fragmented market. The BBO information regime represents a scenario in which price quotes displayed at one venue are not visible to traders acting at another venue. Traders’ orders in this scenario still compete for the best prices across venues, however, as exchanges reroute orders based on the NBBO to prevent trade-throughs. Our alternative scenario allows agents to access cross-market quotes with minimal or even zero latency. Under this latter scenario, prices differ between the fragmented market configurations and the consolidated-market baseline, but the qualitative properties of returns are not strongly affected. How traders access and act on price information in the real-world system may therefore influence what role market fragmentation has on the aggregate price dynamics.

Congress in 1975 noted that the systems for disseminating consolidated market data would ‘form the heart’ of the NMS [53]. O’hara and Ye in 2011 reported findings “consistent with US markets being a single virtual market with multiple points of entry” [140] (p.459). More recently, Budish, Lee, and Shim [33] describe search and access for the best prices across venues as ‘frictionless’ in the modern market. Information access is not literally frictionless in the NMS, however; geography, technological

differences, and the structure of the NMS itself impose constraints and produce heterogeneity in information [8, 173]. ABMs provide a computational tool through which we can explore whether these structural features are sufficient to generate aggregate properties [74].

The model introduced by Wah and Wellman [189] and built on here captures some of these constraints, allowing us to investigate their incremental effects on aggregate price dynamics. By integrating trader logic and parameters from Vyetrenko et al. [187] and the ABIDES platform [36], we introduce a more realistic representation of time to the model. This allows us to more directly compare our experimental settings (e.g., latency) and simulated behavior to empirical observations from the NMS. We find positive support of some empirical properties of returns for each of our baseline, consolidated-market configurations of the model. Namely, we find evidence consistent with the expected return properties of heavy tails, aggregational Gaussianity, and volatility clustering. These results provide partial empirical support for the modeled data-generating process as a representation of the real-world process [196], while caveats to these results and findings contrary to other empirical benchmarks highlight limits to how directly the model’s results should be interpreted for the referent system. We discuss these limitations in more detail below, from which we suggest possible avenues for future research.

4.6.1 LIMITATIONS

Several limitations should be kept in mind when interpreting our results and their applicability to the referent system. We begin by discussing our integration of the Vyetrenko et al. [187] trading strategies into the model. While we provide evidence

this introduced improved empirical relevance for the model in at least some regards, we also find evidence of limitations from this integration. It is important to first highlight that our alignment exercise (Section 4.4.4) showed partial but not perfect alignment. We find similar qualitative relationships between the results from the *base* and *diverse* agent configurations and similar qualitative behavior compared to the empirical benchmarks provided by Vyetrenko et al. The range and exact shape of certain results differ from the original study, however. We know that our market environment is not identical to the ABIDES platform, even in the single-exchange configuration of our model. We specifically only attempt to adopt their trading strategy logic and agent configurations while maintaining a trading environment based on Wah and Wellman’s model [189]. Differences in the trading environment may be driving the noted differences in simulated behavior between our model and theirs. Alternatively, some unintended difference could have been introduced while translating their logic into our model.

In their original study, Vyetrenko et al. [187] also suggest limitations in the empirical validity of the agent configurations we have adopted here. By aiming to align our consolidated-market configurations to the experimental results from Vyetrenko et al., and by choosing to use the random fundamental value instead of historical stock prices, we inherit these limitations and potentially introduce new ones due to the differences between our model settings. For example, we find relatively small price moves and persistently positive linear autocorrelation of returns generated under the *diverse2x* agent configuration for the consolidated market and NBBO information settings. It is possible the liquidity provided by the market maker unrealistically constrains prices to narrow bands under these settings. We find evidence that frag-

mentation (especially under the BBO information setting) helps alleviate this issue, producing less persistent linear autocorrelation and wider price movement. These unrealistic features of the baseline market configuration should be kept in mind when extrapolating from these results, though.

As discussed above, the information scenarios we examine in our model represent two extreme possibilities: a shared view of cross-market information through the NBBO (possibly delayed by SIP latency) or a fully fragmented information setting in which traders only observe the price quotes at their local exchange. We only model communication latency for the SIP, treating all other events as effectively instantaneous. This is consistent with the canonical Wah and Wellman model [189] but not with real markets. We also simulate at most two exchanges, far fewer than the number of national stock exchanges active in the NMS, let alone off-exchange venues. Real-world information scenarios are therefore much more complex, with traders accessing different combinations of data feeds, each received and processed with different amounts of latency.

Finally, some behavior in the model is contrary to expectations from our empirical results in Chapter 2. In the baseline consolidated-market results and the fragmented results under the NBBO information regime, we find patterns of persistent autocorrelation in the 1-minute returns, especially for the *diverse2x* agent configuration. This is in contrast to our empirical results of linear autocorrelation in the NMS at this timescale. The volume/volatility correlation results are relatively weak compared to our empirical results; in many settings the synchronous correlation is even negative, in contrast with the expected relationship. We also find evidence that the BBO information regime may produce tail behavior that is outside empirical estimates for the

base2x agent configuration. More generally, under the BBO information regime fragmentation can generate very large swings and extended periods of volatility. These limitations do not overturn our main comparative findings, but they should moderate strong claims about empirical validity.

4.6.2 FUTURE WORK

These limitations and gaps point to directions for future work. Exploration of the parameter space for the agent strategy settings could improve its empirical validity without otherwise changing the model. Under an indirect calibration approach, fragmentation experiments similar to those examined in this study could be conducted for all parameter settings consistent with the empirical benchmarks [196]. Incorporating communication latency features from the ABIDES platform [36] could help close remaining gaps in our current alignment with the Vyetrenko et al. [187] results while also incorporating additional structural detail based on communication in real-world markets. Alternatively, extending the ABIDES platform to conduct experiments with the structural features of fragmentation examined here would allow for our findings to be independently tested in a similar but different simulation setting.

While we have illuminated some possible effects of fragmentation on price dynamics in this study, our model still significantly simplifies the structure of the NMS. We largely stick to the microstructural features represented by the canonical Wah and Wellman model [189]. Other models have subsequently examined additional features of modern markets, including circuit breakers [183, 174], other forms of latency [183, 38], additional exchanges [183, 66], dark pools [135], and many more. Just as we have shown that fragmentation can affect price dynamics, each of the structural compo-

nents we abstract or omit can and likely does have at least some effect on market dynamics. Large gaps remain between real-world market structure and structure represented in market models. By incrementally incorporating these features into our models and thoroughly testing their effects, we can gain an improved understanding of the mechanisms underlying market behavior and better guide future market design.

CHAPTER 5

CONCLUSION

In this dissertation, we have presented three works that contribute toward addressing the broader question of how aggregate price dynamics in a financial market relate to its underlying organization. We conclude by briefly discussing how the cumulative approach taken in each of these studies can help advance a more robust understanding of this and similar phenomena. We also connect back to ideas from Simon and Schelling, whose quotes we began with in our introduction.

In each of the works presented here, we sought to replicate a prior result before extending its insights towards a specific context—in our case, price dynamics in the modern NMS. In each of these cases, directly building from existing work allowed us to develop more sophisticated frameworks for studying behavior. Testing how our results align with past findings provides a baseline against which we can assess the fidelity of our interpretation and investigate differences, illuminating nuance and possible avenues for future study. This cumulative approach speaks directly to modeling challenges recently emphasized by the SEC, including questions of accuracy, sensitivity, validation, verification, and scalability [180]. In the preceding chapters, we

thoroughly test a set of empirical benchmarks that models are commonly validated against, attempt to verify an existing model through independent replication, and test the sensitivity of modeled behavior to structural and information assumptions.

Simon [163] tells us that scientific knowledge is organized in levels because patterns of behavior at a given level of a system are most clearly discerned by abstracting away the detail of levels far below. Despite marked shifts in the underlying organization of the NMS this century, we find continued evidence in support of recurring properties of returns previously documented by Cont [54] in 2001. In the process, we provide a statistical framework for testing these properties on intraday returns. Application of a consistent framework for measuring statistical properties of returns across different assets, time periods, and markets could help illuminate how uniformly these properties are (or are not) exhibited across different contexts and establish quantitative ranges of empirical behavior. At the same time, extension, critique, and refinement of the tools used to measure these properties can over time provide improved qualitative and quantitative descriptions of this behavior. Similarly improving our understanding of regularities at lower levels of the system could help close the gap between the micro- and macro-levels of the system by documenting the patterns of behavior closer to the micro-level [89, 187].

Schelling [155] highlights that in situations in which people's behavior depends on the behavior of others (e.g., in a financial market), we often must look at the system of interaction between individuals and their environment in order to connect actions at the micro-level to macrobehavior. ABMs provide one means of this type of examination by representing and simulating this system of interactions and observing the resulting behavior. In Chapters 3 and 4, replication, extension, and alignment of ex-

isting financial ABMs allowed us to investigate the incremental effects of fragmented structure on price dynamics while building on the foundation established by previous studies. Through this cumulative process, we were able to investigate how changes in the system of interaction between agents and the market environment affects aggregate behavior. In particular, we found that changes to how agents perceive price quotes in a market can materially affect resulting price dynamics.

For regulators, practitioners, and researchers aiming to understand the effects of interventions at the microstructural level of the market, the behavioral features of complex systems highlighted by Simon and Schelling can provide a challenge: if various settings can produce similar behavior, how can we know whether a model is informative to the context we care about, even if it does produce the desired aggregate behavior? Epstein [74] documents one possible means of addressing this: when presented with multiple candidate microspecifications which generate a macro-level regularity of interest, further work is needed at lower levels of the system to assess which is most empirically tenable. Through the cumulative development of empirical benchmarks and models of behavior at multiple levels of the system, we can develop an improved understanding of the system, more effective interventions, and more informative predictions about real-world scenarios.

BIBLIOGRAPHY

- [1] Anat R Admati and Paul Pfleiderer. “A theory of intraday patterns: Volume and price variability”. In: *The review of financial studies* 1.1 (1988). Publisher: Oxford University Press, pp. 3–40.
- [2] Yacine Aït-Sahalia, Jianqing Fan, and Yingying Li. “The leverage effect puzzle: Disentangling sources of bias at high frequency”. en. In: *Journal of Financial Economics* 109.1 (July 2013), pp. 224–249. ISSN: 0304405X. DOI: [10.1016/j.jfineco.2013.02.018](https://doi.org/10.1016/j.jfineco.2013.02.018). URL: <https://linkinghub.elsevier.com/retrieve/pii/S0304405X13000615>.
- [3] Jeff Alstott, Ed Bullmore, and Dietmar Plenz. “powerlaw: a Python package for analysis of heavy-tailed distributions”. In: *PloS one* 9.1 (2014), e85777.
- [4] Torben G Andersen and Tim Bollerslev. “Intraday periodicity and volatility persistence in financial markets”. In: *Journal of empirical finance* 4.2-3 (1997), pp. 115–158.
- [5] Torben G Andersen et al. “The distribution of realized stock return volatility”. In: *Journal of financial economics* 61.1 (2001), pp. 43–76.
- [6] Torben G. Andersen and Tim Bollerslev, eds. *Volatility*. The International Library of Critical Writings in Economics. Cheltenham, UK: Edward Elgar Publishing, June 2018. ISBN: 978-1-78811-061-7.
- [7] Robert M Anderson et al. “Autocorrelation and partial price adjustment”. In: *Journal of Empirical Finance* 24 (2013), pp. 78–93.
- [8] James J Angel. “When finance meets physics: The impact of the speed of light on financial markets and their regulation”. In: *Financial Review* 49.2 (2014), pp. 271–281.
- [9] James J Angel, Lawrence E Harris, and Chester S Spatt. “Equity trading in the 21st century”. In: *The Quarterly Journal of Finance* 1.01 (2011), pp. 1–53.
- [10] Alain Arneodo, J-F Muzy, and Didier Sornette. “”Direct” causal cascade in the stock market”. In: *The European Physical Journal B-Condensed Matter and Complex Systems* 2 (1998), pp. 277–282.

- [11] Robert Axelrod. “Advancing the art of simulation in the social sciences”. In: *Simulating social phenomena*. Springer, 1997, pp. 21–40.
- [12] Robert Axtell. “Three distinct kinds of empirically-relevant agent-based models”. In: *Brookings Institution Center on Social and Economic Dynamics Working Paper* (2005).
- [13] Robert Axtell et al. “Aligning simulation models: A case study and results”. en. In: *Computational & Mathematical Organization Theory* 1.2 (Feb. 1996), pp. 123–141. ISSN: 1572-9346. DOI: [10.1007/BF01299065](https://doi.org/10.1007/BF01299065). URL: <https://doi.org/10.1007/BF01299065>.
- [14] Robert L Axtell and J Doyne Farmer. “Agent-based modeling in economics and finance: Past, present, and future”. In: *Journal of Economic Literature* (2022).
- [15] Robert L. Axtell. *Why Agents? On the Varied Motivations for Agent Computing in the Social Sciences*. CSED Working Paper 17. The Brookings Institution, Nov. 2000. URL: <https://www.brookings.edu/articles/why-agents-on-the-varied-motivations-for-agent-computing-in-the-social-sciences/>.
- [16] Louis Bachelier. “Théorie de la spéculation”. In: *Annales scientifiques de l’École normale supérieure*. Vol. 17. 1900, pp. 21–86.
- [17] Emmanuel Bacry, Iacopo Mastromatteo, and Jean-François Muzy. “Hawkes processes in finance”. In: *Market Microstructure and Liquidity* 1.01 (2015). Publisher: World Scientific, p. 1550005.
- [18] Markus Baldauf and Joshua Mollner. “High-frequency trading and market performance”. In: *The Journal of Finance* 75.3 (2020). Publisher: Wiley Online Library, pp. 1495–1526.
- [19] Robert P Bartlett III and Justin McCrary. “How rigged are stock markets? Evidence from microsecond timestamps”. In: *Journal of Financial Markets* 45 (2019), pp. 37–60.
- [20] Peter Belcak, Jan-Peter Calliess, and Stefan Zohren. “Fast agent-based simulation framework with applications to reinforcement learning and the study of trading latency effects”. In: *International Workshop on Multi-Agent Systems and Agent-Based Simulation*. Springer, 2021, pp. 42–56.
- [21] Fischer Black and Myron Scholes. “The pricing of options and corporate liabilities”. In: *Journal of political economy* 81.3 (1973). Publisher: The University of Chicago Press, pp. 637–654.
- [22] Pierre Blanc, Jonathan Donier, and J-P Bouchaud. “Quadratic Hawkes processes for financial prices”. In: *Quantitative Finance* 17.2 (2017), pp. 171–188.

- [23] Tim Bollerslev, Ray Y Chou, and Kenneth F Kroner. “ARCH modeling in finance: A review of the theory and empirical evidence”. In: *Journal of econometrics* 52.1-2 (1992), pp. 5–59.
- [24] Richard Bookstaber, Michael D. Foley, and Brian F. Tivnan. “Toward an understanding of market resilience: market liquidity and heterogeneity in the investor decision cycle”. en. In: *Journal of Economic Interaction and Coordination* 11.2 (Oct. 2016), pp. 205–227. ISSN: 1860-7128. DOI: **10.1007/s11403-015-0162-8**. URL: **<https://doi.org/10.1007/s11403-015-0162-8>**.
- [25] Richard Bookstaber, Mark Paddrik, and Brian Tivnan. “An agent-based model for financial vulnerability”. en. In: *Journal of Economic Interaction and Coordination* 13.2 (July 2018), pp. 433–466. ISSN: 1860-7128. DOI: **10.1007/s11403-017-0188-1**. URL: **<https://doi.org/10.1007/s11403-017-0188-1>**.
- [26] Giulio Bottazzi, Giovanni Dosi, and Igor Rebesco. “Institutional architectures and behavioral ecologies in the dynamics of financial markets”. In: *Journal of Mathematical Economics* 41.1-2 (2005), pp. 197–228.
- [27] Jean-Philippe Bouchaud, Andrew Matacz, and Marc Potters. “Leverage Effect in Financial Markets: The Retarded Volatility Model”. In: *Physical Review Letters* 87.22 (Nov. 2001), p. 228701. DOI: **10.1103/PhysRevLett.87.228701**. URL: **<https://link.aps.org/doi/10.1103/PhysRevLett.87.228701>**.
- [28] Jean-Philippe Bouchaud and Marc Potters. *Theory of financial risk and derivative pricing: from statistical physics to risk management*. Cambridge: Cambridge university press, 2003.
- [29] Clive G Bowsher. “Modelling security market events in continuous time: Intensity based, multivariate point process models”. In: *Journal of Econometrics* 141.2 (2007). Publisher: Elsevier, pp. 876–912.
- [30] Anna D Broido and Aaron Clauset. “Scale-free networks are rare”. In: *Nature communications* 10.1 (2019). Publisher: Nature Publishing Group UK London, p. 1017.
- [31] Michael Brolley and David A Cimon. “Order-flow segmentation, liquidity, and price discovery: The role of latency delays”. In: *Journal of Financial and Quantitative Analysis* 55.8 (2020). Publisher: Cambridge University Press, pp. 2555–2587.
- [32] Eric Budish, Peter Cramton, and John Shim. “The high-frequency trading arms race: Frequent batch auctions as a market design response”. In: *The Quarterly Journal of Economics* 130.4 (2015). Publisher: MIT Press, pp. 1547–1621.

- [33] Eric Budish, Robin S Lee, and John J Shim. “A Theory of Stock Exchange Competition and Innovation: Will the Market Fix the Market?” In: *Journal of Political Economy* 132.4 (2024). Publisher: The University of Chicago Press Chicago, IL, pp. 1209–1246.
- [34] Leonard E Burman, W Robert Reed, and James Alm. “A call for replication studies”. In: *Public Finance Review* 38.6 (2010). Publisher: SAGE Publications Sage CA: Los Angeles, CA, pp. 787–793.
- [35] David Byrd. “Explaining agent-based financial market simulation”. In: *arXiv preprint arXiv:1909.11650* (2019).
- [36] David Byrd, Maria Hybinette, and Tucker Hybinette Balch. “Abides: Towards high-fidelity market simulation for ai research”. In: *arXiv preprint arXiv:1904.12066* (2019).
- [37] David Byrd and Antigoni Polychroniadou. “Differentially private secure multi-party computation for federated learning in financial applications”. In: *Proceedings of the first ACM international conference on AI in finance*. 2020, pp. 1–9.
- [38] David Byrd et al. “The importance of low latency to order book imbalance trading strategies”. In: *arXiv preprint arXiv:2006.08682* (2020).
- [39] Ben-Alexander Cassell and Michael P Wellman. “EGTAOnline: An experiment manager for simulation-based game studies”. In: *International Workshop on Multi-Agent Systems and Agent-Based Simulation*. Springer, 2012, pp. 85–100.
- [40] Cboe Global Markets. *U.S. Equities Market Statistics*. July 2026. URL: <https://web.archive.org/web/20260731135115/https://www.cboe.com/markets/us/equities/market-statistics>.
- [41] Anirban Chakraborti et al. “Econophysics review: I. Empirical facts”. en. In: *Quantitative Finance* 11.7 (July 2011), pp. 991–1012. ISSN: 1469-7688, 1469-7696. DOI: [10.1080/14697688.2010.539248](https://doi.org/10.1080/14697688.2010.539248). URL: <http://www.tandfonline.com/doi/abs/10.1080/14697688.2010.539248>.
- [42] Anirban Chakraborti et al. “Econophysics review: II. Agent-based models”. In: *Quantitative Finance* 11.7 (2011), pp. 1013–1041.
- [43] Kalok Chan, Y Peter Chung, and Herb Johnson. “The intraday behavior of bid-ask spreads for NYSE stocks and CBOE options”. In: *Journal of Financial and Quantitative Analysis* 30.3 (1995). Publisher: Cambridge University Press, pp. 329–346.
- [44] Shu-Heng Chen, Chia-Ling Chang, and Ye-Rong Du. “Agent-based economic models and econometrics”. In: *The Knowledge Engineering Review* 27.2 (2012), pp. 187–219.

- [45] Carl Chiarella and Giulia Iori. “A simulation analysis of the microstructure of double auction markets”. In: *Quantitative Finance* 2.5 (Oct. 2002), pp. 346–353. ISSN: 1469-7688. DOI: [10.1088/1469-7688/2/5/303](https://doi.org/10.1088/1469-7688/2/5/303). URL: <https://www.tandfonline.com/doi/abs/10.1088/1469-7688/2/5/303>.
- [46] Chicago Stock Exchange, Inc. *Comment Letter to the U.S. Securities and Exchange Commission Regarding SR-CHX-2016-16 (CHX Liquidity Taking Access Delay)*. Published: Comment letter to Brent J. Fields, Secretary, U.S. Securities and Exchange Commission. Oct. 2016. URL: <https://www.sec.gov/comments/sr-chx-2016-16/chx201616-14.pdf> (visited on 12/14/2025).
- [47] Rémy Chicheportiche and Jean-Philippe Bouchaud. “The fine-structure of volatility feedback I: Multi-scale self-reflexivity”. In: *Physica A: Statistical Mechanics and its Applications* 410 (2014). Publisher: Elsevier, pp. 174–195.
- [48] Peter K Clark. “A subordinated stochastic process model with finite variance for speculative prices”. In: *Econometrica: journal of the Econometric Society* (1973), pp. 135–155.
- [49] Aaron Clauset, Cosma Rohilla Shalizi, and M. E. J. Newman. “Power-Law Distributions in Empirical Data”. In: *SIAM Review* 51.4 (2009), pp. 661–703. ISSN: 0036-1445. URL: <https://www.jstor.org/stable/25662336>.
- [50] Dave Cliff. “An open-source limit-order-book exchange for teaching and research”. In: *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2018, pp. 1853–1860.
- [51] Dave Cliff and Janet Bruten. “Zero is not enough: on the lower limit of agent intelligence for continuous double auction markets”. en. In: *Hewlett-Packard Labs Technical Reports (1997)* (1997).
- [52] Andrew Collins et al. “A call to arms: standards for agent-based modeling and simulation”. In: *Journal of Artificial Societies and Social Simulation* 18.3 (2015), p. 12.
- [53] U. S. Securities and Exchange Commission. “Regulation NMS”. 2023. URL: <https://www.sec.gov/files/rules/final/34-51808.pdf>.
- [54] R. Cont. “Empirical properties of asset returns: stylized facts and statistical issues”. In: *Quantitative Finance* 1.2 (2001), pp. 223–236. DOI: [10.1080/713665670](https://doi.org/10.1080/713665670). URL: <https://doi.org/10.1080/713665670>.
- [55] Rama Cont. “Scaling and correlation in financial data”. <https://arxiv.org/abs/cond-mat/9705075>, May 1997. URL: <http://arxiv.org/abs/cond-mat/9705075>.

- [56] Rama Cont. “Volatility Clustering in Financial Markets: Empirical Facts and Agent-Based Models”. en. In: *Long Memory in Economics*. Ed. by Gilles Teyssière and Alan P. Kirman. Berlin, Heidelberg: Springer, 2007, pp. 289–309. ISBN: 978-3-540-34625-8. DOI: [10.1007/978-3-540-34625-8_10](https://doi.org/10.1007/978-3-540-34625-8_10). URL: https://doi.org/10.1007/978-3-540-34625-8_10.
- [57] Rama Cont and Jean-Philippe Bouchaud. “Herd behavior and aggregate fluctuations in financial markets”. In: *Macroeconomic dynamics* 4.2 (2000). Publisher: Cambridge University Press, pp. 170–196.
- [58] Rama Cont, Marc Potters, and Jean-Philippe Bouchaud. “Scaling in Stock Market Data: Stable Laws and Beyond”. en. In: *Scale Invariance and Beyond*. Ed. by B. Dubrulle, F. Graner, and D. Sornette. Centre de Physique des Houches. Berlin, Heidelberg: Springer, 1997, pp. 75–85. ISBN: 978-3-662-09799-1. DOI: [10.1007/978-3-662-09799-1_5](https://doi.org/10.1007/978-3-662-09799-1_5).
- [59] CTA Plan. *CTA SIP Metrics*. 2026. URL: <https://www.ctaplan.com/metrics>.
- [60] Michael M Dacorogna et al. “A geographical model for the daily and weekly seasonal volatility in the foreign exchange market”. In: *Journal of International Money and Finance* 12.4 (1993), pp. 413–438.
- [61] Marcus G Daniels et al. “Quantitative Model of Price Diffusion and Market Friction Based on Trading as a Mechanistic Random Process”. In: *Physical review letters* 90.10 (2003). Publisher: APS, p. 108102.
- [62] Marco De Luca. *Open Exchange (OpEx) / Wiki / Home*. 2012. URL: <https://sourceforge.net/p/open-exchange/wiki/Home/> (visited on 11/24/2025).
- [63] Lawrence T DeCarlo. “On the meaning and use of kurtosis”. In: *Psychological methods* 2.3 (1997), p. 292.
- [64] Shengwei Ding, John Hanna, and Terrence Hendershott. “How slow is the NBBO? A comparison with direct exchange feeds”. In: *Financial Review* 49.2 (2014), pp. 313–332.
- [65] Philip van Doorn. “Here’s how all 30 Dow Jones Industrial Average stocks performed as the benchmark set another record”. EN-US. Section: Investing. <https://www.marketwatch.com/story/heres-how-all-30-dow-jones-industrial-average-stocks-performed-as-the-benchmark-set-another-record-2018-09-21>, Sept. 2018. URL: <https://www.marketwatch.com/story/heres-how-all-30-dow-jones-industrial-average-stocks-performed-as-the-benchmark-set-another-record-2018-09-21>.

- [66] Matthew Duffin and John Cartlidge. “Agent-based model exploration of latency arbitrage in fragmented financial markets”. In: *2018 IEEE Symposium Series on Computational Intelligence (SSCI)*. IEEE, 2018, pp. 2312–2320.
- [67] Bruce Edmonds and David Hales. “Replication, replication and replication: Some hard lessons from model alignment”. In: *Journal of Artificial Societies and Social Simulation* 6.4 (2003).
- [68] Norman Ehrentreich. *Agent-based modeling: The Santa Fe Institute artificial stock market model revisited*. Springer, 2008.
- [69] Albert Einstein. “On the electrodynamics of moving bodies”. In: *Annalen der physik* 17.10 (1905), pp. 891–921.
- [70] Omar El Euch et al. “The Zumbach effect under rough Heston”. In: *Quantitative finance* 20.2 (2020). Publisher: Taylor & Francis, pp. 235–241.
- [71] Robert F Engle. “Autoregressive conditional heteroscedasticity with estimates of the variance of United Kingdom inflation”. In: *Econometrica: Journal of the econometric society* (1982), pp. 987–1007.
- [72] Robert F Engle. “The econometrics of ultra-high-frequency data”. In: *Econometrica* 68.1 (2000). Publisher: Wiley Online Library, pp. 1–22.
- [73] Robert F Engle and Jeffrey R Russell. “Autoregressive conditional duration: a new model for irregularly spaced transaction data”. In: *Econometrica* (1998). Publisher: JSTOR, pp. 1127–1162.
- [74] Joshua M Epstein. “Agent-based computational models and generative social science”. In: *Complexity* 4.5 (1999), pp. 41–60.
- [75] Eugene F. Fama. “Efficient Capital Markets: A Review of Theory and Empirical Work”. In: *The Journal of Finance* 25.2 (1970), pp. 383–417. ISSN: 0022-1082. DOI: [10.2307/2325486](https://doi.org/10.2307/2325486). URL: <https://www.jstor.org/stable/2325486>.
- [76] J. Doyne Farmer, Paolo Patelli, and Ilija I. Zovko. “The predictive power of zero intelligence in financial markets”. In: *Proceedings of the National Academy of Sciences* 102.6 (Feb. 2005), pp. 2254–2259. DOI: [10.1073/pnas.0409157102](https://doi.org/10.1073/pnas.0409157102). URL: <https://www.pnas.org/doi/abs/10.1073/pnas.0409157102>.
- [77] Stephen Figlewski. “Risk-neutral densities: A review”. In: *Annual Review of Financial Economics* 10.1 (2018). Publisher: Annual Reviews, pp. 329–359.
- [78] Vladimir Filimonov and Didier Sornette. “Power law scaling and “Dragon-Kings” in distributions of intraday financial drawdowns”. In: *Chaos, Solitons & Fractals* 74 (2015). Publisher: Elsevier, pp. 27–45.

- [79] FINRA. “Alternative Display Facility (ADF) \textbar FINRA.org”. 2023. URL: <https://www.finra.org/filing-reporting/alternative-display-facility-adf>.
- [80] FINRA. “OTC Trading Participant Data”. <https://otctransparency.finra.org/OtcData>, Oct. 2023. URL: <https://otctransparency.finra.org/OtcData>.
- [81] FINRA. “Trade Reporting Frequently Asked Questions”. <https://www.finra.org/filing-reporting/market-transparency-reporting/trade-reporting-faq>, Aug. 2024. URL: <https://www.finra.org/filing-reporting/market-transparency-reporting/trade-reporting-faq>.
- [82] FINRA. “Trading Participant Data - ATS Transparency Data”. <https://otctransparency.finra.org/AtsData>, Oct. 2023. URL: <https://otctransparency.finra.org/AtsData>.
- [83] Mark B Garman. “Market microstructure”. In: *Journal of financial Economics* 3.3 (1976), pp. 257–275.
- [84] Ramazan Gencay, Faruk Selcuk, and Brandon Whitcher. *Information flow between volatilities across time scales*. MPRA Paper 10355. University Library of Munich, Germany, Oct. 2004. DOI: None. URL: <https://mpra.ub.uni-muenchen.de/10355/>.
- [85] François Ghoulmie, Rama Cont, and Jean-Pierre Nadal. “Heterogeneity and feedback in an agent-based market model”. en. In: *Journal of Physics: Condensed Matter* 17.14 (Mar. 2005), S1259. ISSN: 0953-8984. DOI: **10.1088/0953-8984/17/14/015**. URL: <https://dx.doi.org/10.1088/0953-8984/17/14/015>.
- [86] Dhananjay K Gode and Shyam Sunder. “Allocative efficiency of markets with zero-intelligence traders: Market as a partial substitute for individual rationality”. In: *Journal of political economy* 101.1 (1993), pp. 119–137.
- [87] Ronald L Goettler, Christine A Parlour, and Uday Rajan. “Informed traders and limit order markets”. In: *Journal of Financial Economics* 93.1 (2009). Publisher: Elsevier, pp. 67–87.
- [88] P. Gopikrishnan et al. “Inverse cubic law for the distribution of stock price variations”. In: *The European Physical Journal B - Condensed Matter and Complex Systems* 3.2 (May 1998), pp. 139–140. ISSN: 1434-6036. DOI: **10.1007/s100510050292**. URL: <https://doi.org/10.1007/s100510050292>.
- [89] Martin D Gould et al. “Limit order books”. In: *Quantitative Finance* 13.11 (2013). Publisher: Taylor & Francis, pp. 1709–1742.
- [90] Volker Grimm et al. “A standard protocol for describing individual-based and agent-based models”. In: *Ecological modelling* 198.1-2 (2006), pp. 115–126.

- [91] Volker Grimm et al. “The ODD protocol: a review and first update”. In: *Ecological modelling* 221.23 (2010), pp. 2760–2768.
- [92] Volker Grimm et al. “The ODD protocol: a review and first update”. In: *Ecological modelling* 221.23 (2010). Publisher: Elsevier, pp. 2760–2768.
- [93] Volker Grimm et al. “Using the ODD protocol and NetLogo to replicate agent-based models”. In: *Ecological Modelling* 501 (2025). Publisher: Elsevier, p. 110967.
- [94] Jonas Hauke, Sebastian Achter, and Matthias Meyer. “Theory Development Via Replicated Simulations and the Added Value of Standards”. In: *Journal of Artificial Societies and Social Simulation* 23.1 (2020).
- [95] Höskuldur Ari Hauksson et al. “Multivariate extremes, aggregation and risk estimation”. In: *Quantitative Finance* 1 (2001), pp. 79–95.
- [96] Friedrich A. Hayek. “The Use of Knowledge in Society”. In: *The American Economic Review* 35.4 (Sept. 1945), pp. 519–530.
- [97] Brian Heath, Raymond Hill, and Frank Ciarallo. “A survey of agent-based modeling practices (January 1998 to July 2008)”. In: *Journal of Artificial Societies and Social Simulation* 12.4 (2009), p. 9.
- [98] Bruce M. Hill. “A Simple General Approach to Inference About the Tail of a Distribution”. In: *The Annals of Statistics* 3.5 (1975). Publisher: Institute of Mathematical Statistics, pp. 1163–1174. ISSN: 00905364, 21688966. URL: <http://www.jstor.org/stable/2958370> (visited on 07/14/2025).
- [99] Cars H Hommes. “Modeling the stylized facts in finance through simple non-linear adaptive systems”. In: *Proceedings of the National Academy of Sciences* 99.suppl_3 (2002), pp. 7221–7228.
- [100] Marat Ibragimov, Rustam Ibragimov, and Paul Kattuman. “Emerging markets and heavy tails”. In: *Journal of Banking & Finance* 37.7 (2013). Publisher: Elsevier, pp. 2546–2559.
- [101] Giulia Iori. “A microsimulation of traders activity in the stock market: the role of heterogeneity, agents’ interactions and trade frictions”. In: *Journal of Economic Behavior & Organization* 49.2 (2002), pp. 269–285.
- [102] J.P. Morgan/Reuters. *RiskMetrics—Technical Document*. Tech. rep. Edition: 4. New York: J.P. Morgan/Reuters, 1996.
- [103] Prem C Jain and Gun-Ho Joh. “The dependence between hoURLy prices and trading volume”. In: *Journal of Financial and Quantitative Analysis* 23.3 (1988), pp. 269–283.

- [104] Mogens H. Jensen, Anders Johansen, and Ingve Simonsen. “Inverse statistics in economics: the gain–loss asymmetry”. en. In: *Physica A: Statistical Mechanics and its Applications*. Proceedings of the International Econophysics Conference 324.1 (June 2003), pp. 338–343. ISSN: 0378-4371. DOI: [10.1016/S0378-4371\(02\)01884-8](https://doi.org/10.1016/S0378-4371(02)01884-8). URL: <https://www.sciencedirect.com/science/article/pii/S0378437102018848>.
- [105] Anders Johansen, Ingve Simonsen, and Mogens H Jensen. “Optimal investment horizons for stocks and markets”. In: *Physica A: Statistical Mechanics and its Applications* 370.1 (2006), pp. 64–67.
- [106] Neil Johnson et al. “Abrupt rise of new machine ecology beyond human response time”. en. In: *Scientific Reports* 3.1 (Sept. 2013), p. 2627. ISSN: 2045-2322. DOI: [10.1038/srep02627](https://doi.org/10.1038/srep02627). URL: <https://www.nature.com/articles/srep02627>.
- [107] Paul E Johnson. “Agent-based modeling: What I learned from the artificial stock market”. In: *Social Science Computer Review* 20.2 (2002). Publisher: Sage Publications Sage CA: Thousand Oaks, CA, pp. 174–186.
- [108] Kei Katahira et al. “Development of an agent-based speculation game for higher reproducibility of financial stylized facts”. In: *Physica A: Statistical Mechanics and its Applications* 524 (2019), pp. 503–518.
- [109] Andrei Kirilenko et al. “The flash crash: High-frequency trading in an electronic market”. In: *The Journal of Finance* 72.3 (2017), pp. 967–998.
- [110] Christopher G Lamoureux and William D Lastrapes. “Heteroskedasticity in stock return data: Volume versus GARCH effects”. In: *The journal of finance* 45.1 (1990), pp. 221–229.
- [111] Blake LeBaron. “A builder’s guide to agent-based financial markets”. In: *Quantitative finance* 1.2 (2001), p. 254.
- [112] Blake LeBaron. “Agent-based financial markets: Matching stylized facts with style”. In: *Post Walrasian Macroeconomics: Beyond the DSGE Model* 221 (2006), p. 235.
- [113] Blake LeBaron. “Building the Santa Fe artificial stock market”. In: *Physica A* 1 (2002), p. 20.
- [114] Michael Lewis. *Flash boys: a Wall Street revolt*. WW Norton & Company, 2014.
- [115] Yanhui Liu, Parameswaran Gopikrishnan, H Eugene Stanley, et al. “Statistical properties of the volatility of price fluctuations”. In: *Physical review e* 60.2 (1999), p. 1390.

- [116] Yanhui Liu et al. “Correlations in economic time series”. In: *Physica A: Statistical Mechanics and its Applications* 245.3-4 (1997). Publisher: Elsevier, pp. 437–440.
- [117] Michael A Lombardi. “Synchronizing stock market clocks to UTC (NIST)”. In: *2020 XXXIIIrd General Assembly and Scientific Symposium of the International Union of Radio Science*. <https://ieeexplore.ieee.org/abstract/document/9232016>: IEEE, 2020, pp. 1–4.
- [118] Thomas Lux and Michele Marchesi. “Scaling and criticality in a stochastic multi-agent model of a financial market”. In: *Nature* 397.6719 (1999), pp. 498–500.
- [119] Thomas Lux and Michele Marchesi. “Volatility clustering in financial markets: a microsimulation of interacting agents”. In: *International journal of theoretical and applied finance* 3.04 (2000), pp. 675–702.
- [120] Thomas Lux and Remco CJ Zwinkels. “Empirical validation of agent-based models”. In: *Handbook of computational economics*. Vol. 4. Elsevier, 2018, pp. 437–488.
- [121] Yiping Ma et al. “Flamingo: Multi-round single-server secure aggregation with applications to private federated learning”. In: *2023 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2023, pp. 477–496.
- [122] Michael Macy and Yoshimichi Sato. “The surprising success of a replication that failed”. In: *Journal of Artificial Societies and Social Simulation* 13.2 (2010), p. 9.
- [123] Yannick Malevergne, Vladilen Pisarenko, and Didier Sornette. “Empirical Distributions of Stock Returns: Between the Stretched Exponential and the Power Law?” In: *Quantitative Finance* 5 (Aug. 2005), pp. 379–401. DOI: **10.1080/14697680500151343**.
- [124] Benoit Mandelbrot et al. “The variation of certain speculative prices”. In: *Journal of business* 36.4 (1963), p. 394.
- [125] Rosario N. Mantegna and H. Eugene Stanley. *Introduction to Econophysics: Correlations and Complexity in Finance*. Cambridge, UK: Cambridge University Press, 1999. DOI: **10.1017/CB09780511755767**.
- [126] Chris Mascioli et al. “A Financial Market Simulation Environment for Trading Agents Using Deep Reinforcement Learning”. In: *Proceedings of the 5th ACM International Conference on AI in Finance*. 2024, pp. 117–125.

- [127] Frank McGroarty et al. “High frequency trading strategies, market fragility and price spikes: an agent based model perspective”. In: *Annals of Operations Research* 282.1 (Nov. 2019), pp. 217–244. ISSN: 1572-9338. DOI: [10.1007/s10479-018-3019-4](https://doi.org/10.1007/s10479-018-3019-4). URL: <https://doi.org/10.1007/s10479-018-3019-4>.
- [128] Albert J Menkveld. “The economics of high-frequency trading: Taking stock”. In: *Annual Review of Financial Economics* 8.1 (2016), pp. 1–24.
- [129] Ugo Merlone, Michele Sonnessa, Pietro Terna, et al. “Horizontal and vertical multiple implementations in a model of industrial districts”. In: *JASSS* 11 (2008), pp. 1–25.
- [130] Robert C Merton. “Theory of rational option pricing”. In: *The Bell Journal of economics and management science* (1973). Publisher: JSTOR, pp. 141–183.
- [131] Thomas Mikosch and Cătălin Stărică. “Nonstationarities in Financial Time Series, the Long-Range Dependence, and the IGARCH Effects”. In: *The Review of Economics and Statistics* 86.1 (2004). Publisher: The MIT Press, pp. 378–390. ISSN: 00346535, 15309142. URL: <http://www.jstor.org/stable/3211680> (visited on 07/14/2025).
- [132] Suchismita Mishra and Le Zhao. “Order Routing Decisions for a Fragmented Market: A Review”. In: *Journal of Risk and Financial Management* 14.11 (2021), p. 556.
- [133] Takanobu Mizuta et al. “Do dark pools stabilize markets and reduce market impacts? investigations using multi-agent simulations”. In: *2014 IEEE Conference on computational intelligence for financial engineering & economics (CIFEr)*. IEEE, 2014, pp. 71–76.
- [134] Takanobu Mizuta et al. “Effects of dark pools on financial markets’ efficiency and price discovery function: an investigation by multi-agent simulations”. In: *Evolutionary and Institutional Economics Review* 12 (2015), pp. 375–394.
- [135] Sheung Yin Kevin Mo, Mark Paddrik, and Steve Y Yang. “A study of dark pool trading using an agent-based model”. In: *2013 IEEE Conference on Computational Intelligence for Financial Engineering & Economics (CIFEr)*. IEEE, 2013, pp. 19–26.
- [136] Claudia E. Moise and Paca Flaherty. “*Limit Up–Limit Down*” Pilot Plan and Associated Events. White Paper. U.S. Securities, Exchange Commission, Division of Economic, and Risk Analysis, Mar. 2017. URL: <https://www.sec.gov/files/dera-luld-white-paper.pdf>.
- [137] Ulrich A Müller et al. “Statistical study of foreign exchange rates, empirical evidence of a price change scaling law, and intraday analysis”. In: *Journal of Banking & Finance* 14.6 (1990), pp. 1189–1208.

- [138] Ulrich A. Müller et al. “Volatilities of different time resolutions — Analyzing the dynamics of market components”. en. In: *Journal of Empirical Finance*. High Frequency Data in Finance, Part 1 4.2 (June 1997), pp. 213–239. ISSN: 0927-5398. DOI: [10.1016/S0927-5398\(97\)00007-8](https://doi.org/10.1016/S0927-5398(97)00007-8). URL: <https://www.sciencedirect.com/science/article/pii/S0927539897000078>.
- [139] Jean-François Muzy, Jean Delour, and Emmanuel Bacry. “Modelling fluctuations of financial time series: from cascade process to stochastic volatility model”. In: *The European Physical Journal B-Condensed Matter and Complex Systems* 17 (2000), pp. 537–548.
- [140] Maureen O’Hara and Mao Ye. “Is market fragmentation harming market quality?” In: *Journal of Financial Economics* 100.3 (2011), pp. 459–474.
- [141] Maureen O’Hara. “High frequency market microstructure”. In: *Journal of financial economics* 116.2 (2015), pp. 257–270.
- [142] Maury FM Osborne. “Brownian motion in the stock market”. In: *Operations research* 7.2 (1959), pp. 145–173.
- [143] Adrian Pagan. “The econometrics of financial markets”. In: *Journal of empirical finance* 3.1 (1996), pp. 15–102.
- [144] Richard G Palmer et al. “An artificial stock market”. In: *Artificial Life and Robotics* 3.1 (1999). Publisher: Springer, pp. 27–31.
- [145] Vasiliki Plerou et al. “Economic fluctuations and anomalous diffusion”. In: *Physical Review E* 62.3 (2000), R3023.
- [146] Gary Polhill and Luis Izquierdo. “Lessons learned from converting the artificial stock market to interval arithmetic”. In: *Journal of Artificial Societies and Social Simulation* 8.2 (2005).
- [147] J Gary Polhill, Luis R Izquierdo, and Nicholas M Gotts. “The ghost in the model (and other effects of floating point arithmetic)”. In: *Journal of Artificial Societies and Social Simulation* 8.1 (2005).
- [148] Marc Potters, Rama Cont, and Jean-Philippe Bouchaud. “Financial markets as adaptive systems”. In: *EPL* 41 (1998), pp. 239–244.
- [149] T. Preis et al. “Multi-agent-based Order Book Model of financial markets”. en. In: *Europhysics Letters* 75.3 (July 2006), p. 510. ISSN: 0295-5075. DOI: [10.1209/epl/i2006-10139-0](https://doi.org/10.1209/epl/i2006-10139-0). URL: <https://iopscience.iop.org/article/10.1209/epl/i2006-10139-0/meta>.
- [150] Marco Raberto et al. “Agent-based simulation of a financial market”. In: *Physica A: Statistical Mechanics and its Applications* 299.1-2 (2001), pp. 319–327.

- [151] Wolfgang Radax and Bernhard Rengs. “Prospects and pitfalls of statistical testing: Insights from replicating the demographic prisoner’s dilemma”. In: *Journal of Artificial Societies and Social Simulation* 13.4 (2010), p. 1.
- [152] Ethan Ratliff-Crain et al. “Revisiting Stylized Facts for Modern Stock Markets”. In: *2023 IEEE International Conference on Big Data (BigData)*. Sorrento, Italy, 2023, pp. 1814–1823. DOI: **10.1109/BigData59044.2023.10386957**.
- [153] Riccardo Rebonato. *Volatility and correlation: the perfect hedger and the fox*. John Wiley & Sons, 2005.
- [154] Paul A. Samuelson. “Proof That Properly Anticipated Prices Fluctuate Randomly”. In: *Industrial Management Review* 6.2 (1965), pp. 41–49.
- [155] Thomas C. Schelling. *Micromotives and Macrobehavior*. Paperback. New York: W. W. Norton & Company, 1978. ISBN: 978-0-393-09010-9.
- [156] Chad W Seagren. “A replication and analysis of tiebout competition using an agent-based computational model”. In: *Social Science Computer Review* 33.2 (2015). Publisher: SAGE Publications Sage CA: Los Angeles, CA, pp. 198–216.
- [157] Securities and Exchange Commission. *Self-Regulatory Organizations; Chicago Stock Exchange, Inc.; Notice of Filing of Proposed Rule Change to Adopt the CHX Liquidity Enhancing Access Delay*. Securities Exchange Act Release 34-80041. U.S. Securities and Exchange Commission, Feb. 2017. URL: <https://www.sec.gov/files/rules/sro/chx/2017/34-80041.pdf> (visited on 12/14/2025).
- [158] Securities and Exchange Commission. *Self-Regulatory Organizations; Chicago Stock Exchange, Inc.; Notice of Filing of Proposed Rule Change to Adopt the CHX Liquidity Taking Access Delay*. Securities Exchange Act Release 34-78860. U.S. Securities and Exchange Commission, Sept. 2016. URL: <https://www.sec.gov/files/rules/sro/chx/2016/34-78860.pdf> (visited on 12/14/2025).
- [159] Securities and Exchange Commission. *Self-Regulatory Organizations; The Options Clearing Corporation; Notice of Filing of Proposed Rule Change by The Options Clearing Corporation Concerning Modifications to its By-Laws and Rules Primarily to Discontinue Certain Outmoded or Unused Products and Services*. Sept. 2024. URL: <https://www.sec.gov/files/rules/sro/occ/2024/34-101189.pdf>.

- [160] Moonis Shakeel and Bhavana Srivastava. “Stylized facts of high-frequency financial time series data”. In: *Global Business Review* 22.2 (2021), pp. 550–564.
- [161] A Christian Silva and Victor M Yakovenko. “Stochastic volatility of financial markets as the fluctuating rate of trading: An empirical study”. In: *Physica A: Statistical Mechanics and its Applications* 382.1 (2007), pp. 278–285.
- [162] Herbert A Simon. “A behavioral model of rational choice”. In: *The quarterly journal of economics* (1955), pp. 99–118.
- [163] Herbert A Simon. “The organization of complex systems”. In: *Models of discovery: And other topics in the methods of science*. Springer, 1977, pp. 245–261.
- [164] Herbert A Simon and Albert Ando. “Aggregation of variables in dynamic systems”. In: *Econometrica: journal of the Econometric Society* (1961), pp. 111–138.
- [165] Herbert A. Simon. “The architecture of complexity”. In: *Proceedings of the American Philosophical Society* 106.6 (1962), pp. 467–482.
- [166] Johannes Siven, Jeffrey Lins, and Jonas Lundbek Hansen. “A multiscale view on inverse statistics and gain/loss asymmetry in financial time series”. In: *Journal of Statistical Mechanics: Theory and Experiment* 2009.02 (2009), P02004.
- [167] Eric Smith et al. “Statistical theory of the continuous double auction”. In: *Quantitative finance* 3.6 (2003). Publisher: IOP Publishing, p. 481.
- [168] Thomas Stöckl, Jürgen Huber, and Michael Kirchler. “Bubble measures in experimental asset markets”. In: *Experimental Economics* 13.3 (2010), pp. 284–298.
- [169] Leonna Szangolies et al. “Visual ODD: A standardised visualisation illustrating the narrative of agent-based models”. In: *Journal of Artificial Societies and Social Simulation* 27.4 (2024). Publisher: JASSS.
- [170] Texas Stock Exchange. *Member Readiness and Launch Guide*. 2026. URL: <https://www.txse.com/trading-membership/member-readiness-and-launch-guide>.
- [171] Jan C Thiele and Volker Grimm. “Replicating and breaking models: good for you and good for ecology”. In: *Oikos* 124.6 (2015). Publisher: Wiley Online Library, pp. 691–696.
- [172] Brian Tivnan et al. “Adding to the Regulator’s Toolbox: Integration and Extension of Two Leading Market Models”. en. In: *Papers* (May 2011). URL: <https://ideas.repec.org/p/arx/papers/1105.5439.html>.

- [173] Brian F. Tivnan et al. “Fragmentation and inefficiencies in US equity markets: Evidence from the Dow 30”. en. In: *PLOS ONE* 15.1 (Jan. 2020), e0226968. ISSN: 1932-6203. DOI: **10.1371/journal.pone.0226968**. URL: **<https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0226968>**.
- [174] Brian F. Tivnan et al. “Towards a Model of the National Market System: Fragmented and Heterogenous Venues”. In: *Proceedings of the 2020 Conference of The Computational Social Science Society of the Americas*. Ed. by Zining Yang and Elizabeth von Briesen. Cham: Springer International Publishing, 2021, pp. 265–279. ISBN: 978-3-030-83418-0.
- [175] Brian F. Tivnan et al. “Towards a model of the U.S. stock market: How important is the securities information processor?” In: *2017 Winter Simulation Conference (WSC)*. Las Vegas, NV, Dec. 2017, pp. 1181–1192. DOI: **10.1109/WSC.2017.8247865**.
- [176] Lora Todorova and Bodo Vogt. “Power law distribution in high frequency financial data? An econometric analysis”. en. In: *Physica A: Statistical Mechanics and its Applications* 390.23 (Nov. 2011), pp. 4433–4444. ISSN: 0378-4371. DOI: **10.1016/j.physa.2011.07.035**. URL: **<https://www.sciencedirect.com/science/article/pii/S0378437111005826>**.
- [177] U.S. Securities and Exchange Commission. *Common Stock*. Publication Title: Glossary. Feb. 2024. URL: **<https://www.sec.gov/resources-small-businesses/glossary#CS>**.
- [178] U.S. Securities and Exchange Commission. *Disclosure of Order Execution Information*. Published: Final Rule, Release No. 34-99679, File No. S7-29-22. Mar. 2024.
- [179] U.S. Securities and Exchange Commission. *Regulation NMS*. Tech. rep. Release No. 34-51808. U.S. Securities and Exchange Commission, 2005. URL: **<https://www.sec.gov/files/rules/final/34-51808.pdf>** (visited on 10/20/2023).
- [180] U.S. Securities and Exchange Commission. *Regulation NMS: Minimum Pricing Increments, Access Fees, and Transparency of Better Priced Orders*. Tech. rep. Release No. 34-101070. U.S. Securities and Exchange Commission, 2024. URL: **<https://www.sec.gov/files/rules/final/2024/34-101070.pdf>**.
- [181] US U.S. Securities and Exchange Commission. “Equity market structure literature review part i: Market fragmentation”. In: *Securities and Exchange Commision Working Paper* (2013).

- [182] UTP Plan. *UTP SIP Metrics*. 2026. URL: <https://www.utpplan.com/metrics>.
- [183] Colin M Van Oort et al. “Adaptive Agents and Data Quality in Agent-Based Financial Markets”. In: *arXiv preprint arXiv:2311.15974* (2023).
- [184] Colin M. Van Oort et al. “Ecological and Coevolutionary Dynamics in Modern Markets Yield Nonstationarity in Market Efficiencies”. In: *Complexity* 2022.1 (2022). _eprint: <https://onlinelibrary.wiley.com/doi/pdf/10.1155/2022/7109529>, p. 7109529. DOI: <https://doi.org/10.1155/2022/7109529>. URL: <https://onlinelibrary.wiley.com/doi/abs/10.1155/2022/7109529>.
- [185] Wouter Vermeer et al. “Leveraging modularity during replication of high-fidelity models: lessons from replicating an agent-based model for HIV prevention”. In: *Journal of artificial societies and social simulation: JASSS* 23.4 (2020), p. 7.
- [186] Christian E Vincenot. “How new concepts become universal scientific approaches: insights from citation network analysis of agent-based complex systems science”. In: *Proceedings of the Royal Society B: Biological Sciences* 285.1874 (2018). Publisher: The Royal Society, p. 20172360.
- [187] Svitlana Vyetrenko et al. “Get real: Realism metrics for robust limit order book market simulations”. In: *Proceedings of the First ACM International Conference on AI in Finance*. 2020, pp. 1–8.
- [188] Elaine Wah. “Computational Models of Algorithmic Trading in Financial Markets.” PhD Thesis. 2016.
- [189] Elaine Wah and Michael P Wellman. “Latency arbitrage in fragmented markets: A strategic agent-based analysis”. In: *Algorithmic Finance* 5.3-4 (2016), pp. 69–93.
- [190] Elaine Wah and Michael P Wellman. “Latency arbitrage, market fragmentation, and efficiency: A two-market model”. In: *Proceedings of the fourteenth ACM conference on Electronic commerce*. 2013, pp. 855–872.
- [191] Elaine Wah, Mason Wright, and Michael P Wellman. “Welfare effects of market making in continuous double auctions”. In: *Journal of Artificial Intelligence Research* 59 (2017), pp. 613–650.
- [192] Michael P Wellman, Karl Tuyls, and Amy Greenwald. “Empirical game theoretic analysis: A survey”. In: *Journal of artificial intelligence research* 82 (2025), pp. 1017–1076.
- [193] Uri Wilensky and William Rand. “Making models match: Replicating an agent-based model”. In: *Journal of Artificial Societies and Social Simulation* 10.4 (2007), p. 2.

- [194] Oliver Will. “Resolving a replication that failed: News on the Macy & Sato model”. In: *Journal of Artificial Societies and Social Simulation* 12.4 (2009), p. 11.
- [195] Oliver Will and Rainer Hegselmann. “A Replication That Failed on the Computational Model in Michael W. Macy and Yoshimichi Sato: Trust, Cooperation and Market Formation in the US and Japan. Proceedings of the National Academy of Sciences, May 2002”. In: *Journal of Artificial Societies and Social Simulation* 11.3 (2008), p. 3.
- [196] Paul Windrum, Giorgio Fagiolo, and Alessio Moneta. “Empirical validation of agent-based models: Alternatives and prospects”. In: *Journal of Artificial Societies and Social Simulation* 10.2 (2007), p. 8.
- [197] Robert A Wood, Thomas H McInish, and J Keith Ord. “An investigation of transactions data for NYSE stocks”. In: *The Journal of Finance* 40.3 (1985). Publisher: Wiley Online Library, pp. 723–739.
- [198] Gilles Zumbach and Paul Lynch. “Heterogeneous volatility cascade in financial markets”. In: *Physica A: Statistical Mechanics and its Applications* 298.3-4 (2001). Publisher: Elsevier, pp. 521–529.

APPENDIX A

SUPPLEMENTARY MATERIAL FOR CHAPTER 2

A.1 RESULTS ON RAW VERSUS NORMALIZED RETURNS

As noted in Section 2.1.2, there is a u-shaped daily volatility pattern characteristic to financial markets. Before assessing most of the stylized fact metrics discussed in our main results, we corrected for this effect using two different methods of normalizing by activity level. We standardized the clock-returns by the average activity level at their corresponding time of day, and we looked at the normalized event-time returns, which fixes the number of trades per unit of time. These normalization methods have possible implications for the statistical analyses done in our study and possibly on our results. Using the raw returns in clock-time, we conducted the same tests from our main paper as detailed in Section 2.3. These results are presented side-by-side with

the normalized clock-time results, reproduced from our main paper (Figures A.1, A.2, A.3, A.4, A.5, A.6, and A.7).

Importantly, our high-level takeaways do not change; we found support for the same eight stylized facts with the raw clock-time returns as we saw for the normalized clock-time returns (see Table 2.1). The finer details of some facts are impacted. For example, the correlation between volume and volatility (Figure A.5) is stronger across the symbols in the raw returns and stays positive longer for more lags as we look at past or future volatility. This makes sense, given our clock-time normalization method is correcting for increased return magnitudes at the starts and ends of the day, and volume follows a similar pattern. The volatility clustering results (Figure A.3) have differing ranges of exponent β fit for the power-law decay. The shape of the decay is also not as straight in the raw returns as in the normalized returns. Relative to the normalized return results, the raw return metrics have wider ranges across the symbols in our sample, along with much wider bootstrap confidence intervals indicating greater uncertainty. This uncertainty motivates our focus on the normalized returns in our main results. The choice to focus on the normalized returns is also consistent with much of the literature we draw from in this study (e.g., Blanc, Donier, and Bouchaud [22], Malevergne, Pisarenko, and Sornette [123], Müller et al. [138], Muzy, Delour, and Bacry [139], and Zumbach and Lynch [198]).

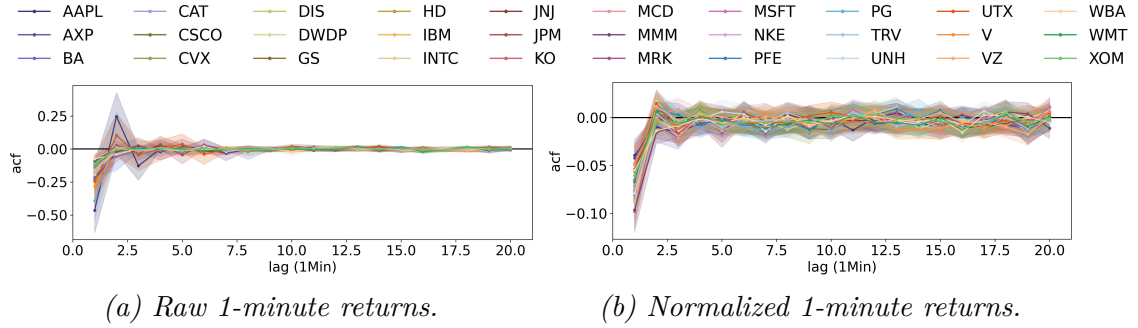


Figure A.1: Linear autocorrelation of returns. Please note, the color legend shown here is used for all result plots and is consistent with the main body results.

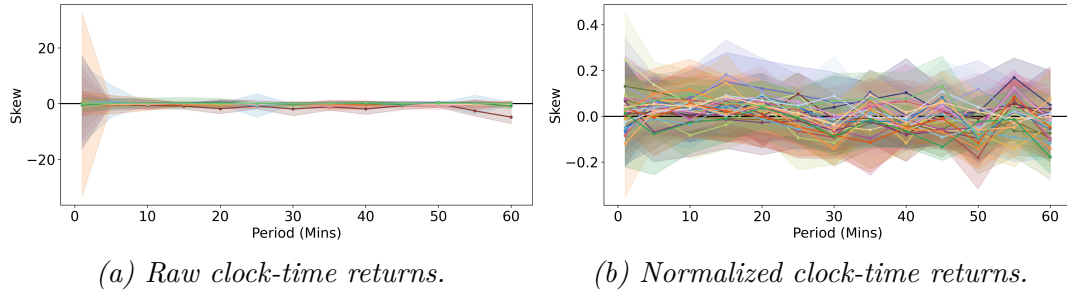


Figure A.2: Skew of the return distributions.

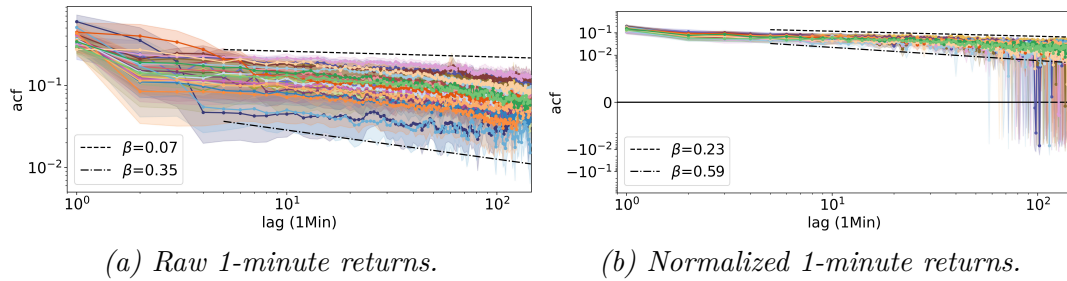


Figure A.3: Autocorrelation of absolute returns.

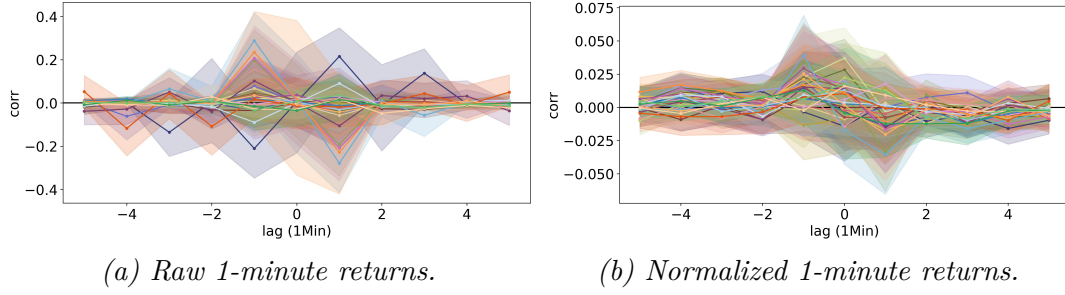


Figure A.4: Leverage effect $L(\tau, \Delta t) = \text{corr}(r(t, \Delta t), |r(t + \tau, \Delta t)|)$ for the raw and normalized returns.

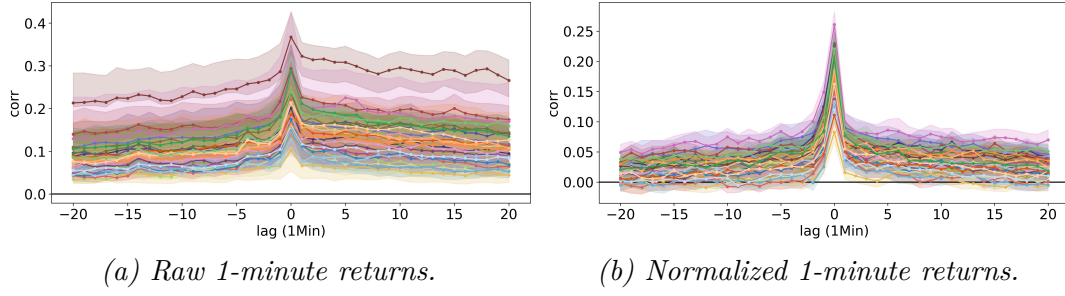


Figure A.5: Share volume versus lagged volatility.

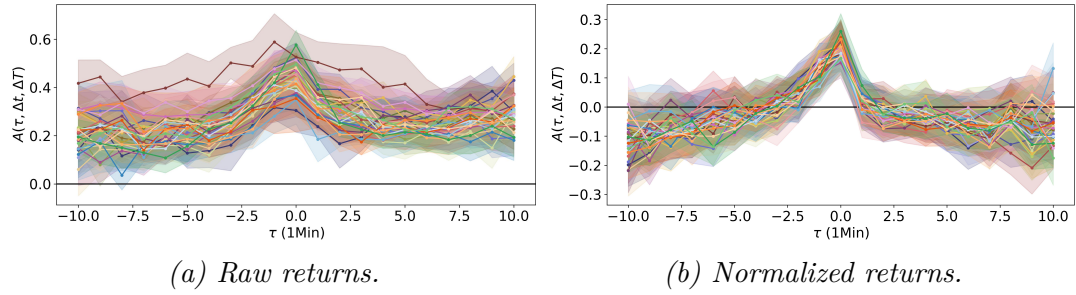


Figure A.6: Average 1-minute volatility versus lagged 30-minute volatility.

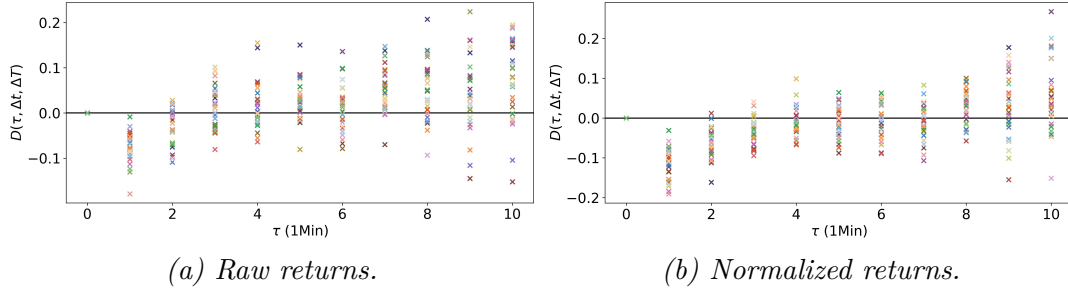
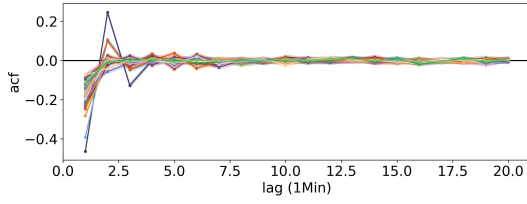


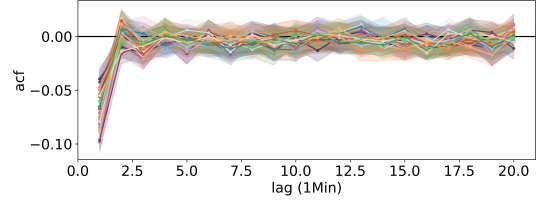
Figure A.7: Asymmetry effect — Difference between positive and negative lags for the correlation between the average 1-minute volatility and lagged 30-minute volatility.

A.2 THEORETICAL VERSUS BOOTSTRAP CONFIDENCE BANDS

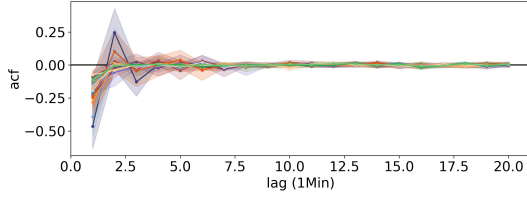
The raw return results illustrate the importance of bootstrapping in estimating uncertainty for the metrics tested here. Figure A.8 compares theoretical confidence bands for linear autocorrelation with bootstrap confidence intervals for both raw and normalized 1-minute returns. The theoretical confidence bands are much narrower than their bootstrap counterparts, failing to capture the additional uncertainty present in the raw return data. For some symbols, the theoretical confidence bands suggest significant nonzero linear autocorrelation at the second lag, whereas the bootstrap confidence intervals for the same data include zero, indicating that the observed autocorrelations may not be statistically significant. After normalizing returns, the theoretical confidence bands align much more closely with the bootstrap intervals, suggesting that normalization mitigates much of the additional variability observed in raw returns.



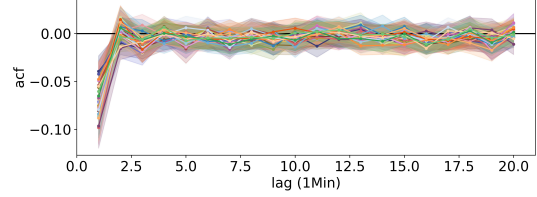
(a) Raw 1-minute returns with theoretical confidence bands.



(b) Normalized 1-minute returns theoretical confidence bands.



(c) Raw 1-minute returns with bootstrap confidence intervals.

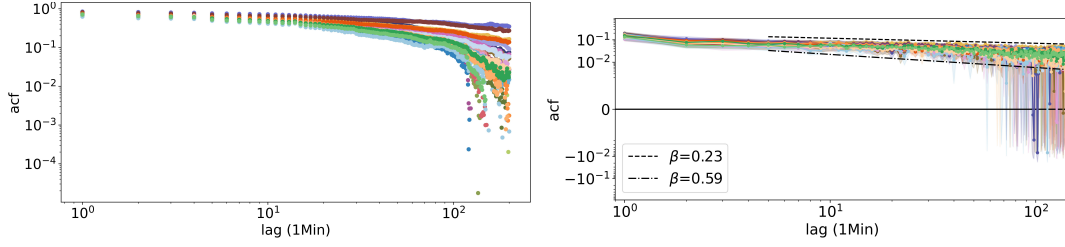


(d) Normalized 1-minute returns with bootstrap confidence intervals.

Figure A.8: Linear autocorrelation of returns with theoretical versus bootstrap 95% confidence intervals.

A.3 AUTOCORRELATION OF TRADING VOLUME

In the main paper, we saw evidence of seven stylized facts exhibited in both the clock- and event-time, with some level of variation between these clocks depending on the specific measure. One possible factor that could influence the relationship between these two clocks (and any difference between them) is clustering of trading behavior in time. Similar to volatility clustering, we find evidence that the volume of trades in a minute clusters in time, and the autocorrelation of trading volume appears to exhibit a slow, power-law decay. This is shown in Figure A.9. This is one avenue of future study for examining the relationship between clock-time and event-time in financial series.



(a) 1-minute autocorrelation of trade volume. (b) 1-minute autocorrelation of absolute normalized returns.

Figure A.9: Side-by-side comparison of the autocorrelation of trade volume and absolute returns at 1-minute timescale.

A.4 VOLUME/VOLATILITY CORRELATION — ADDITIONAL DETAILS

Past results have shown an asymmetry in the volume/volatility correlation relationship where past volatility predicts volume better than the other way around [103]. In Figure A.11, we test whether there is an asymmetry in our lagged volume/volatility correlation results by differencing the correlation values at lags of equal magnitude:

$$D_V(\tau, \Delta t) = V(\tau, \Delta t) - V(-\tau, \Delta t),$$

where

$$V(\tau, \Delta t) = \text{corr}(\text{volume}(t, \Delta(t)), |r(t + \tau, \Delta t)|).$$

There is a slight asymmetry where the correlation between trade volume and lagged volatility is stronger for $\tau = -1$ than for $\tau = 1$. However, the effect is weak (less than 0.04 in magnitude for most symbols) and disappears by the second lag. There

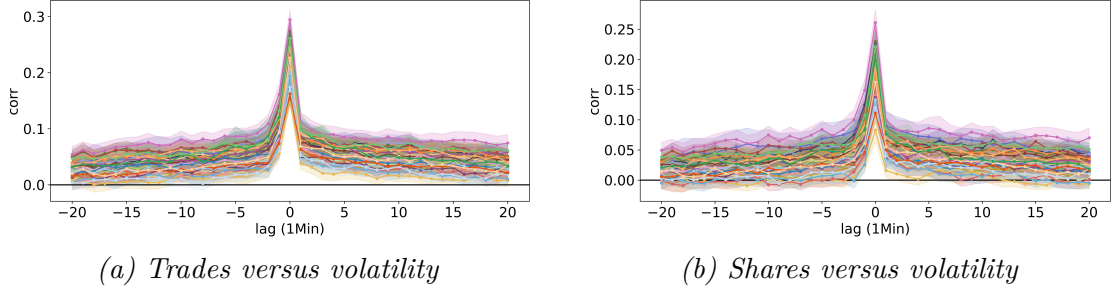


Figure A.10: Volume versus lagged volatility: $V(\tau, \Delta t) = \text{corr}(\text{volume}(t, \Delta(t)), |r(t + \tau, \Delta t)|)$, with $\Delta t = 1$ min. Reproduced from our main results (Figure 2.9).

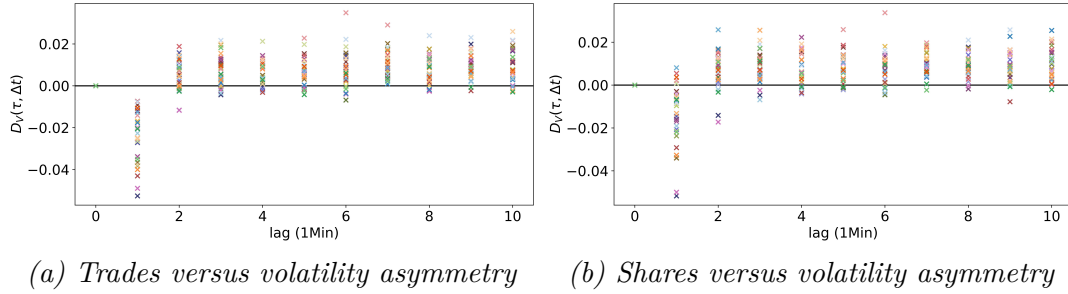


Figure A.11: Asymmetry in volume/volatility correlation: $D_V(\tau, \Delta t) = V(\tau, \Delta t) - V(-\tau, \Delta t)$, with $\Delta t = 1$ min.

is no consistent asymmetry when we use shares as our volume metric. We therefore do not find a clear asymmetry for this metric in the normalized 1-minute results.

A.5 SUPPLEMENTARY RESULTS

In Figure A.12, we show an alternative measure of gain/loss asymmetry (Fact #3). Looking at the percentage of extreme returns that were gains versus losses, we did not see a disproportionate amount of large losses. This runs counter to the claims of Fact #3, as noted in our main results (Section 2.3.3).

In Table A.1, we give an alternative measure of intermittency, looking at the

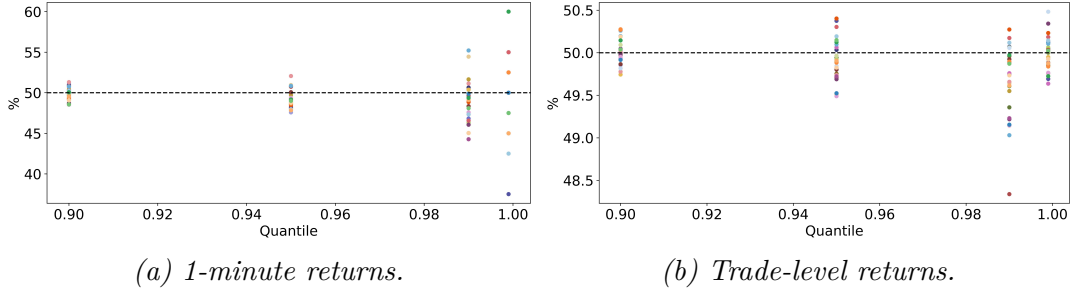


Figure A.12: Percentage of normalized returns which were losses above a quantile-cutoff.

kurtosis of the interarrival times for extreme returns. Specifically looking at the 99th-percentile largest returns, the interarrival times are heavy-tailed, with kurtosis values above zero. This is an indication of intermittency, in addition to the results shown in Section 2.3.5. Some of the symbols have confidence level ranges near zero in the 1-minute results, however. This may be partly due to the small sample size of 99th-percentile returns at this timescale. We tested alternative percentile cutoffs for this metric and the extreme return Fano factor (see Section 2.3.5). These metrics are plotted as a function of percentile cutoff in Figures A.13 and A.14. The plots show the metrics are relatively stable, with a slight decrease as we get further out into the extremes. Further analysis could look into whether this trajectory is due to sample size or if the arrival of large returns become better described by a Poisson process as we get further out into the tails.

In our main paper, we use the absolute returns for each volatility-related metric tested. The leverage effect results by Bouchaud, Matacz, and Potters [27] looked at the relationship between returns and squared returns. To make sure our choice of volatility metric was not the reason we did not find evidence of this effect at the tested timescales, we re-measured the effect using the squared normalized returns as

Symbol	1-min Return	Trade Return
AAPL	2.77 (1.39–4.37)	6906.05 (184.49–10018.69)
AXP	2.58 (0.52–4.58)	211.48 (91.50–371.51)
BA	4.68 (2.17–7.60)	709.77 (407.35–1067.83)
CAT	3.89 (1.95–6.20)	229.46 (132.13–314.50)
CSCO	4.35 (1.44–7.10)	247.69 (132.06–390.45)
CVX	2.92 (1.03–5.05)	376.51 (144.48–656.01)
DIS	3.04 (1.29–5.07)	237.72 (151.08–334.19)
DWDP	4.83 (2.56–7.38)	325.50 (171.07–509.77)
GS	3.46 (0.89–5.91)	222.80 (108.10–342.09)
HD	5.74 (3.16–8.91)	227.06 (135.34–312.52)
IBM	5.34 (2.74–7.83)	348.35 (232.35–478.72)
INTC	2.95 (0.79–4.91)	309.88 (153.11–471.97)
JNJ	4.02 (1.45–6.57)	274.83 (162.81–363.20)
JPM	2.26 (0.83–3.89)	426.94 (124.18–814.57)
KO	5.31 (3.07–7.61)	2442.90 (276.21–3794.80)
MCD	2.66 (1.32–4.26)	251.59 (138.12–355.26)
MMM	3.93 (1.48–6.01)	186.29 (116.74–267.83)
MRK	3.26 (1.37–5.38)	226.30 (121.55–348.22)
MSFT	3.96 (1.73–5.96)	215.91 (131.04–318.41)
NKE	4.01 (1.47–6.28)	675.64 (112.71–1260.87)
PFE	3.82 (1.78–5.87)	234.29 (133.59–360.99)
PG	3.57 (1.96–5.38)	979.22 (202.32–1640.54)
TRV	3.24 (1.74–4.77)	128.97 (88.19–168.48)
UNH	2.30 (1.18–3.59)	327.43 (195.20–485.90)
UTX	4.86 (2.67–7.42)	678.54 (229.87–1119.88)
V	3.98 (2.16–6.17)	608.71 (187.43–1169.27)
VZ	5.73 (1.52–9.15)	282.01 (173.07–395.69)
WBA	5.02 (2.93–7.44)	400.08 (122.79–604.51)
WMT	2.04 (0.79–3.44)	410.09 (130.30–737.32)
XOM	2.82 (1.18–4.66)	111.74 (76.17–158.01)

Table A.1: Kurtosis of the (intraday) interarrival times for 99th-percentile extreme normalized returns. This is an alternative measure of intermittency, as discussed in Section 2.3.5. Values in parentheses give the 95% bootstrap confidence interval ranges.

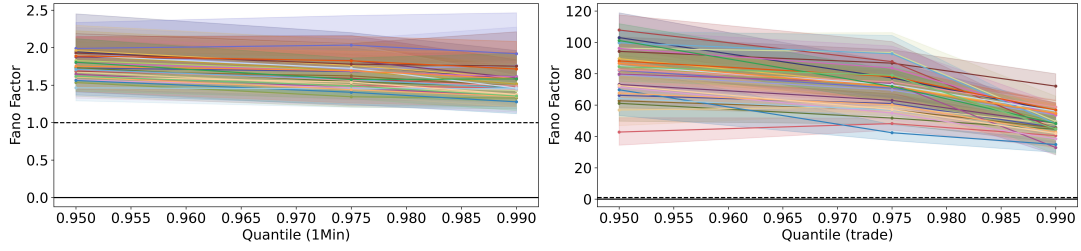


Figure A.13: Fano factor of the number of extreme returns that arrive within a coarse period of time, plotted for different quantile cutoffs. The coarse period for normalized 1-minute returns is 30-minutes, and the coarse period for trade-level returns is 1000 trades. Shaded regions give the 95th-percentile bootstrap confidence interval for the metric.

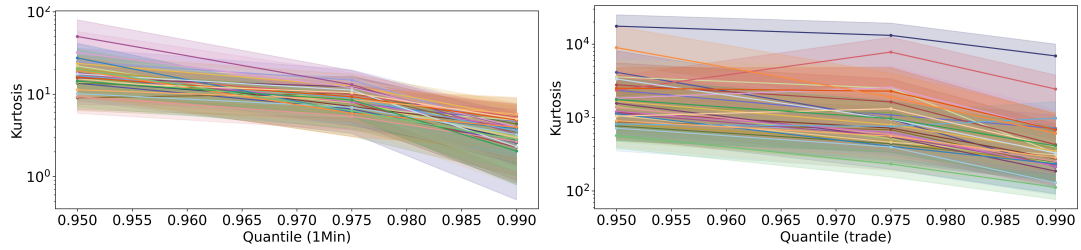


Figure A.14: Kurtosis of the distribution of interarrival times of extreme returns, plotted for different quantile cutoffs. Shaded regions give the 95th-percentile bootstrap confidence interval for the metric.

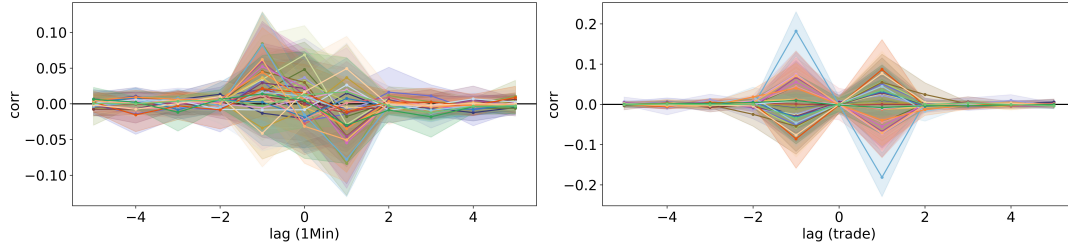


Figure A.15: Leverage effect with squared normalized returns as the volatility measure: $L(\tau, \Delta t) = \text{corr}(r(t, \Delta t), (r(t + \tau, \Delta t))^2)$.

the volatility measure instead. In Figure A.15, we show the relationship between returns and squared returns:

$$L(\tau, \Delta t) = \text{corr}(r(t, \Delta t), (r(t + \tau, \Delta t))^2).$$

Consistent with our main results, the symbols have varying sign and strength for this relationship, in contrast to the expected stylized fact.

A.6 ALTERNATIVE VISUALIZATIONS OF SCALING RELATIONSHIPS

Here we show alternative visualizations of our kurtosis and absolute autocorrelation results in order to better visualize the scaling relationships for these metrics. In our main results, we plot the excess kurtosis as a function of timescale (Figure 2.3), using the below metric:

$$K(\Delta t) = \frac{\langle (r(t, \Delta t) - \langle r(t, \Delta t) \rangle)^4 \rangle}{\sigma(\Delta t)^4} - 3,$$

where a normal distribution will have $K(\Delta t) \approx 0$. Consistent with the property of aggregational Gaussianity, as timescale increased, we found $K(\Delta t)$ approaches zero or even went negative for some stocks, particularly for the normalized clock-time and even-time returns.

In order to better visualize the rate at which kurtosis decreases as a function of timescale, we can plot the raw kurtosis:

$$K^*(\Delta t) = \frac{\langle (r(t, \Delta t) - \langle r(t, \Delta t) \rangle)^4 \rangle}{\sigma(\Delta t)^4}.$$

Equivalently, $K^*(\Delta t) = K(\Delta t) + 3$. The raw kurtosis will always be positive, and a normal distribution will have a raw kurtosis of approximately 3. We plot the raw kurtosis as a function of timescale for the raw clock-time returns (Figure A.16), normalized clock-time returns (Figure A.17), and the normalized event-time returns ((Figure A.18)). Changes in magnitude have equal scale between the x-axis and y-axis for each result, allowing for the slope of the decay to be visually assessed more easily.

In Section 2.3.4, we plot the absolute autocorrelation of the normalized returns as a function of lag:

$$C^0(\tau, \Delta t) = \text{corr}(|r(t, \Delta t)|, |r(t + \tau, \Delta t)|).$$

According to Cont [54], the measure is expected to decay roughly as a power-law:

$$C^0(\tau, \Delta t) \propto \tau^{-\beta},$$

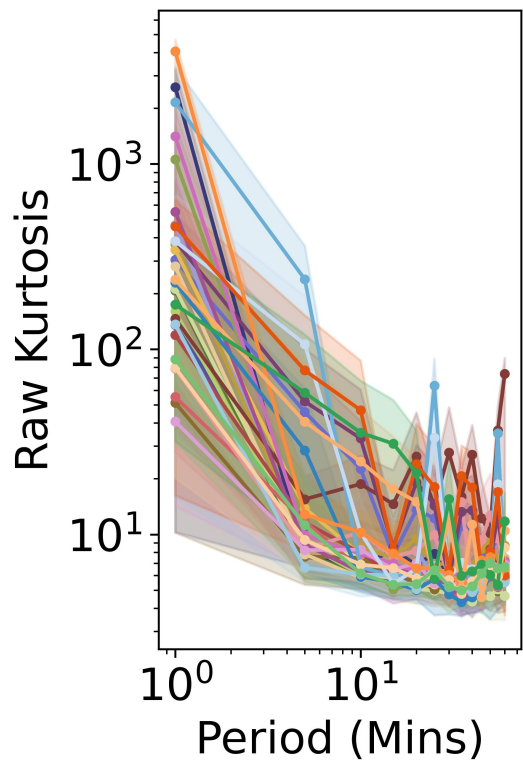


Figure A.16: Raw kurtosis as a function of timescale for the raw clock-time returns. Solid lines show the kurtosis for a symbol's full sample of intraday returns. Shaded regions show the 95% confidence interval from bootstrapping.

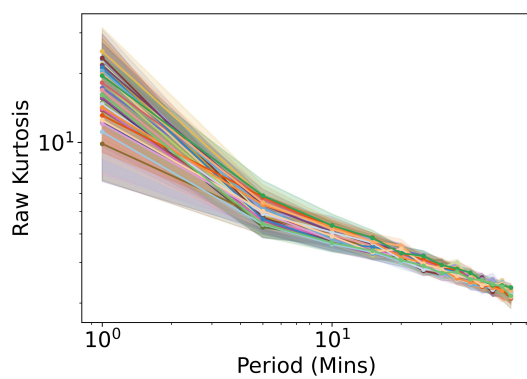


Figure A.17: Raw kurtosis as a function of timescale for the normalized clock-time returns. Solid lines show the kurtosis for a symbol's full sample of intraday returns. Shaded regions show the 95% confidence interval from bootstrapping.

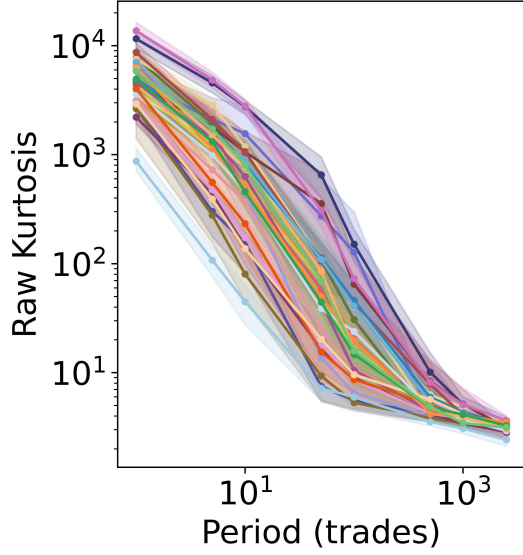


Figure A.18: Raw kurtosis as a function of timescale for the normalized event-time returns. Solid lines show the kurtosis for a symbol's full sample of intraday returns. Shaded regions show the 95% confidence interval from bootstrapping.

with $\beta \in [0.2, 0.4]$. The full set of clock-time results include some negative values at later lags (Figure 2.7). In log-scale, this distorts the y-axis in order to show the positive and negative values in the same plot.

In Figure A.19, we present two alternative views of our absolute autocorrelation results. We first plot the full absolute-autocorrelation results in linear scale. We then remove the non-positive values and plot the results in log-log scale with equal scales for changes in magnitude in the x- and y-axes. Note that the trade-level returns did not have negative values to begin with, so the only change relative to our main results is forcing equal scaling between the x- and y-axis. The linear-scale figures show the asymptotic path of the absolute autocorrelation as a function of lag. The log-scale results without the positive results show the slope of this decay, with guide lines showing the maximum and minimum slopes of power-law fits. The power-law fits

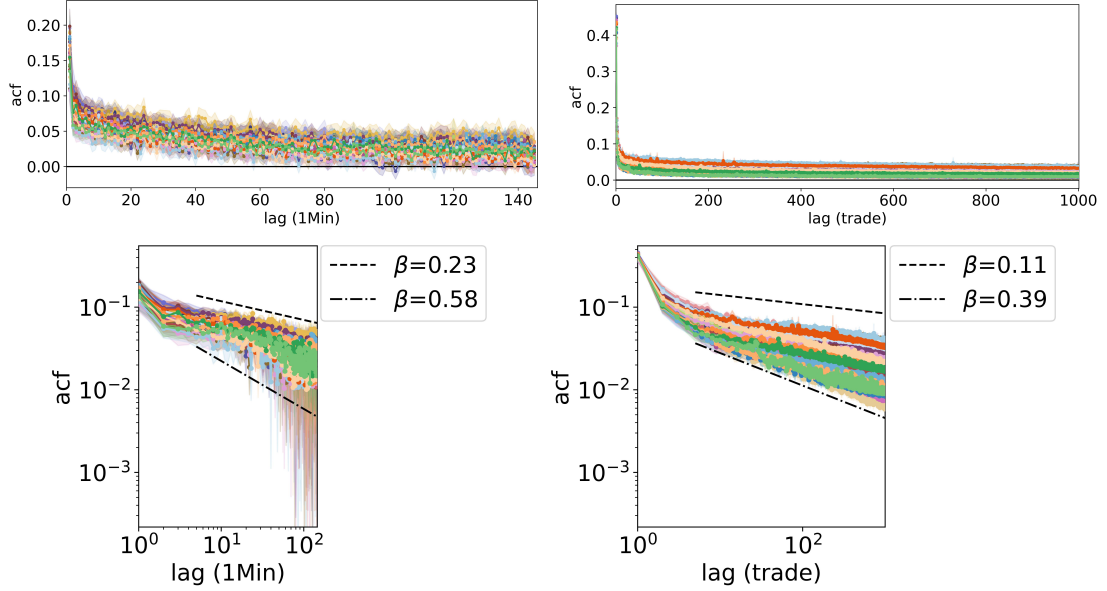
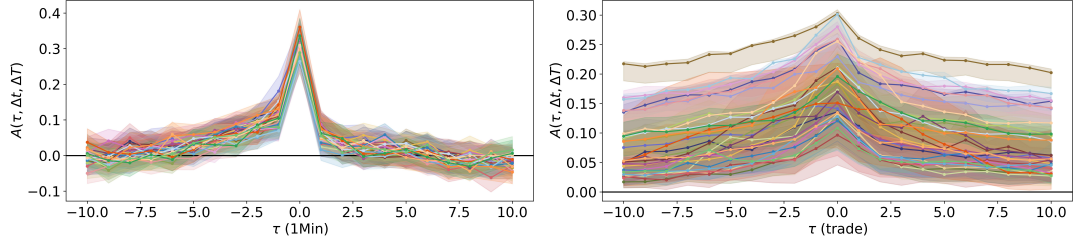


Figure A.19: Alternative visualizations of the autocorrelation of the absolute normalized returns. Top results show the absolute autocorrelation in linear scale. Bottom results show the same results in log-log scale with the non-positive values removed for the clock-time results. Solid lines show the absolute autocorrelation for a symbol's full sample of intraday returns. Shaded regions show the 95% confidence interval from bootstrapping. Dashed lines in the log-log plots give upper and lower bounds of the power-law exponents β fit to the decay trajectories, starting at lag $\tau = 5$. These lines are offset above and below the empirical results for clarity.

are calculated using log-regression on each symbol's absolute autocorrelation results where $\tau \geq 5$ and $C^0(\tau, \Delta t) > 0.01$.

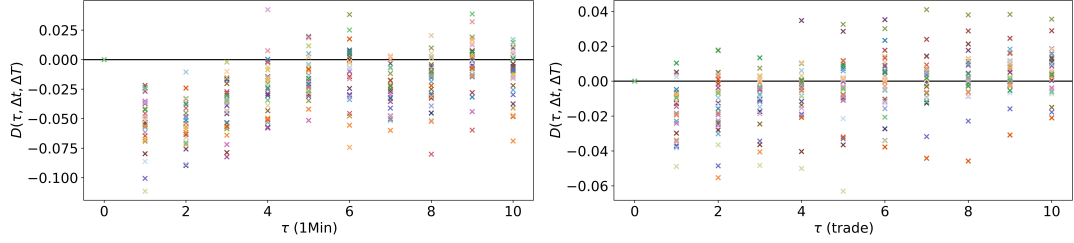
A.7 ASYMMETRY IN TIMESCALES — ALTERNATIVE MEASURES

Fact #11, asymmetry in timescales, expects coarse volatility measures to predict fine volatility better than the other way around. In our main results (Section 2.3.8), we used absolute returns at a coarse timescale ΔT and average absolute returns



(a) 1-minute versus lagged 10-minute volatility. (b) Trade versus lagged 100-trade volatility.

Figure A.20: Asymmetry in timescales.

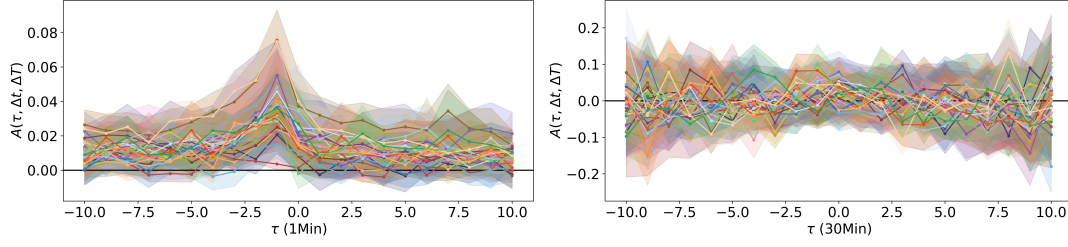


(a) 1-minute versus lagged 10-minute volatility for the normalized clock-time returns. (b) Trade versus lagged 100-trade volatility for the normalized clock-time returns.

Figure A.21: Asymmetry effect — Difference between positive and negative lags for the correlation between the average fine volatility and lagged coarse volatility.

at a shorter timescale Δt to test this effect. We specifically showed the results for $\Delta T = 30Min$ and $\Delta t = 1Min$ in clock-time and $\Delta T = 1000$ trades and $\Delta t = 1$ trade in even-time. In Figure A.20, we show alternative coarse timescales $\Delta T = 10Min$ and $\Delta T = 100$ trades, respectively.

We also considered an alternative fine volatility measure, using Rogers-Satchell volatility as done by Blanc, Donier, and Bouchaud [22]. For time period $[t - \Delta T, t]$, let O_t , C_t , H_t , and L_t denote the open, close, high, and low prices, respectively. We consider the open price to be the price of the first trade in the bucket, the close to be the last, and any bucket that has no trades will forward fill the most recent price



(a) Rogers-Satchell volatility versus lagged 1-minute volatility. (b) Rogers-Satchell volatility versus lagged 30-minute volatility.

Figure A.22: Correlation between Rogers-Satchell volatility and lagged coarse volatility — normalized clock-time returns.

for each. The Rogers-Satchell volatility is then:

$$\sigma^{RS}(t, \Delta T) = \sqrt{\ln(H_t/O_t) \times \ln(H_t/C_t) + \ln(L_t/O_t) \times \ln(L_t/C_t)}.$$

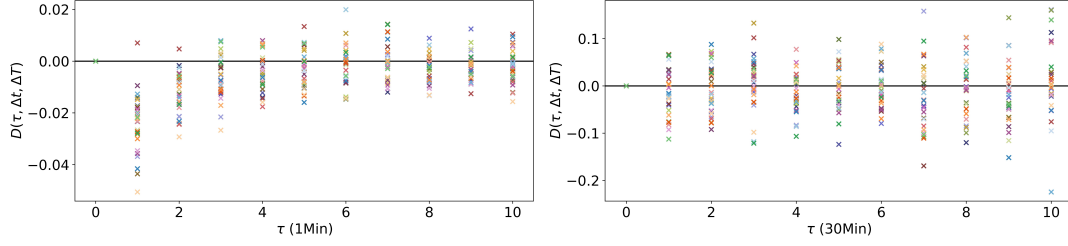
The Rogers-Satchell volatility is normalized by the intraday volatility pattern in the same way as the returns (see Section 2.2). The alternative asymmetry measure is then the difference between positive and negative lags of the below:

$$A_{RS}(\tau, \Delta T) = \text{corr}(\sigma^{RS}(t, \Delta T), |r(T + \tau, \Delta T)|).$$

Our findings, shown in Figure A.22 for clock-time and Figure A.24 for event-time, do not change the main takeaways for this fact discussed in Section 2.3.8.

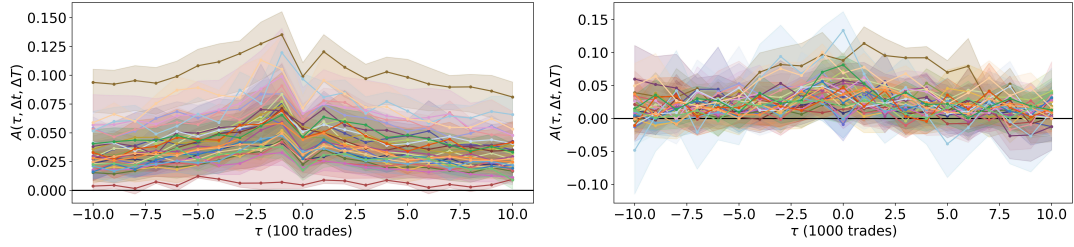
A.8 INTERARRIVAL TIMES

Table A.2 details the intraday interarrival times for trades in our sample.



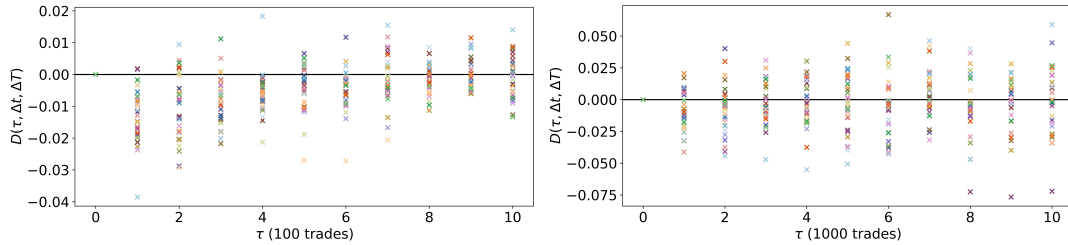
(a) Asymmetry difference in correlation for positive and negative lags of Rogers-Satchell volatility versus lagged 1-minute Satchell volatility. (b) Asymmetry difference in correlation for positive and negative lags of Rogers-Satchell volatility versus lagged 30-minute volatility.

Figure A.23: Asymmetry in timescales using RS-volatility — normalized returns.



(a) Rogers-Satchell volatility versus lagged 100-trade volatility. (b) Rogers-Satchell volatility versus lagged 1000-trade volatility.

Figure A.24: Correlation between Rogers-Satchell volatility and coarse volatility — normalized trade-time.



(a) Asymmetry difference in correlation for positive and negative lags of Rogers-Satchell volatility versus lagged 100-trade Satchell volatility. (b) Asymmetry difference in correlation for positive and negative lags of Rogers-Satchell volatility versus lagged 1000-trade volatility.

Figure A.25: Asymmetry in timescales using RS-volatility — normalized trade-time returns.

symbol	mean	median	std dev	min	max
AAPL	92187.4448	542	253142.7725	0	12571649
AXP	692207.9126	820	1911627.7109	0	119150021
BA	371787.9327	1142	1070198.1843	0	44135765
CAT	459240.8081	650	1322471.0403	0	44879485
CSCO	193369.4276	147	604495.5908	0	25425292
CVX	447459.8725	1010	1113191.8908	0	93435678
DIS	370537.1973	1114	933530.6497	0	57402802
DWDP	326028.1423	336	965076.6627	0	126575915
GS	580411.4588	763	1689139.4617	0	68538772
HD	451671.9537	1036	1132654.7523	0	76497996
IBM	482454.2264	1220	1250915.2628	0	82863601
INTC	167717.2605	141	558131.4384	0	26597742
JNJ	354335.9755	1045	922782.6087	0	71360236
JPM	209847.1484	457	570390.7108	0	22944580
KO	311519.2083	282	935603.4328	0	46698708
MCD	638397.9650	1472	1603954.7510	0	54331818
MMM	852194.4061	1081	2189984.9764	0	146362458
MRK	313920.7991	434	887260.7131	0	66412614
MSFT	99193.0671	207	303788.1717	0	12622029
NKE	483389.8170	537	1358197.5504	0	54277013
PFE	221150.2204	236	689749.7424	0	73738081
PG	335203.7812	444	949010.3113	0	129897333
TRV	1364061.3279	1125	3683287.1008	0	111138471
UNH	543007.1342	1029	1372532.9913	0	87779112
UTX	549267.6523	999	1446871.4256	0	43139194
V	304772.9679	834	803048.7751	0	42872582
VZ	268070.9614	334	781668.6576	0	29908213
WBA	506693.1092	337	1583376.9705	0	70183033
WMT	371879.4968	651	974687.4033	0	61237236
XOM	279634.9969	396	794565.4217	0	78406653
all	421387.12	694.03	1155177.90	0	146362458

Table A.2: Intraday interarrival times (in μs) of trades for Dow 30 stocks in our sample.

A.9 TAIL FITS AND DISTRIBUTIONAL COMPARISONS

Tables A.1, A.1, and A.1 give the detailed clock-time power-law fit and distributional comparison metrics summarized in Section 2.3.2. Tables A.1, A.1, and A.1 give the same for the event-time returns.

symbol	Left					Right					
	$x_{\min}$	$\hat{\alpha}$	$N_{x \geq x_{\min}}$	$PL_{us_LN}(r,p)$	$LN_{us_exp}(r,p)$	$x_{\min}$	$\hat{\alpha}$	$N_{x \geq x_{\min}}$	$PL_{us_LN}(r,p)$	$PL_{us_exp}(r,p)$	$LN_{us_exp}(r,p)$
AAPL	1.85092	4.24 ± 0.19	1123	(-0.11, 0.81)	(59.32, 0.00)	1.96555	4.29 ± 0.21	926	(-0.00, 0.96)	(66.08, 0.00)	(66.08, 0.00)
AXP	2.38074	4.89 ± 0.33	522	(-1.45, 0.02)	(26.18, 0.00)	1.68799	4.11 ± 0.16	1491	(-0.43, 0.61)	(73.99, 0.00)	(74.42, 0.00)
BA	1.89646	4.16 ± 0.19	1049	(-0.40, 0.23)	(70.28, 0.00)	1.84828	4.17 ± 0.19	1105	(0.01, 0.67)	(73.50, 0.00)	(73.49, 0.00)
CAT	1.85032	4.09 ± 0.18	1078	(0.01, 0.88)	(76.43, 0.00)	2.07396	4.37 ± 0.23	853	(-0.06, 0.05)	(74.09, 0.00)	(74.14, 0.00)
CSCO	1.95303	4.56 ± 0.23	920	(-0.47, 0.04)	(74.06, 0.00)	2.22868	4.47 ± 0.26	661	(-0.40, 0.13)	(45.47, 0.00)	(45.88, 0.00)
CVX	1.74518	4.37 ± 0.18	1333	(0.00, 0.97)	(77.30, 0.00)	1.92634	4.4 ± 0.21	1033	(0.01, 0.76)	(59.14, 0.00)	(59.13, 0.00)
DIS	1.65289	3.99 ± 0.15	1462	(0.01, 0.86)	(101.47, 0.00)	1.53831	3.94 ± 0.14	1778	(-0.16, 0.76)	(112.66, 0.00)	(112.81, 0.00)
DWDP	1.65965	4.19 ± 0.17	1430	(-4.57, 0.00)	(117.59, 0.00)	1.82565	3.94 ± 0.17	1162	(0.01, 0.83)	(83.04, 0.00)	(83.03, 0.00)
GS	2.63249	5.05 ± 0.41	383	(0.01, 0.69)	(16.36, 0.02)	2.48444	4.73 ± 0.32	510	(-0.31, 0.66)	(13.49, 0.07)	(13.80, 0.04)
HD	2.15223	4.1 ± 0.22	739	(-0.88, 0.02)	(57.05, 0.00)	1.51086	3.81 ± 0.13	1781	(-0.93, 0.48)	(110.07, 0.00)	(111.00, 0.00)
IBM	1.70413	3.91 ± 0.16	1253	(0.03, 0.25)	(115.20, 0.00)	1.67388	3.9 ± 0.16	1344	(-2.76, 0.04)	(114.51, 0.00)	(117.26, 0.00)
INTC	1.80797	4.18 ± 0.18	1159	(-0.45, 0.01)	(92.38, 0.00)	1.83592	4.28 ± 0.19	1160	(0.03, 0.25)	(91.20, 0.00)	(91.17, 0.00)
JNJ	1.54653	3.78 ± 0.13	1645	(0.02, 0.46)	(142.51, 0.00)	1.73464	3.85 ± 0.16	1234	(0.02, 0.71)	(96.08, 0.00)	(96.06, 0.00)
JPM	2.04019	4.59 ± 0.24	865	(-0.06, 0.30)	(51.74, 0.00)	1.68862	4.16 ± 0.16	1479	(-0.27, 0.69)	(74.46, 0.00)	(74.73, 0.00)
KO	1.85120	4.09 ± 0.19	1065	(-0.64, 0.58)	(85.03, 0.00)	1.66088	4.05 ± 0.15	1503	(-0.13, 0.78)	(88.74, 0.00)	(88.87, 0.00)
MCD	3.00496	4.28 ± 0.39	272	(0.00, 0.96)	(13.34, 0.01)	1.68029	4.07 ± 0.16	1459	(-0.47, 0.61)	(78.02, 0.00)	(78.49, 0.00)
MMM	1.66946	3.94 ± 0.15	1392	(0.01, 0.76)	(106.32, 0.00)	1.84130	4.19 ± 0.19	1107	(0.02, 0.39)	(86.87, 0.00)	(86.85, 0.00)
MRK	1.96267	4.4 ± 0.22	943	(0.01, 0.68)	(56.03, 0.00)	1.90800	4.2 ± 0.19	1068	(0.01, 0.87)	(63.68, 0.00)	(63.68, 0.00)
MSFT	1.67624	4.22 ± 0.16	1529	(-3.24, 0.23)	(49.09, 0.01)	2.36397	4.66 ± 0.3	588	(-0.03, 0.57)	(31.66, 0.00)	(31.70, 0.00)
NKE	1.82766	4.4 ± 0.19	1232	(-2.58, 0.27)	(33.44, 0.03)	2.20538	4.46 ± 0.26	701	(-0.05, 0.87)	(30.62, 0.00)	(30.67, 0.00)
PFE	1.59569	4.05 ± 0.15	1613	(0.01, 0.83)	(121.53, 0.00)	1.80970	4.2 ± 0.18	1205	(-1.02, 0.00)	(97.52, 0.00)	(98.55, 0.00)
PG	1.99506	3.91 ± 0.19	866	(-0.00, 0.98)	(56.80, 0.00)	1.67232	3.85 ± 0.15	1363	(0.00, 0.96)	(98.39, 0.00)	(98.38, 0.00)
TRV	2.70095	4.85 ± 0.41	338	(0.00, 0.72)	(17.99, 0.01)	2.03142	4.29 ± 0.22	888	(-1.17, 0.38)	(25.34, 0.01)	(26.50, 0.00)
UNH	1.93233	4.28 ± 0.2	1005	(-0.01, 0.95)	(54.35, 0.00)	2.21140	4.6 ± 0.26	715	(0.01, 0.72)	(38.79, 0.00)	(38.78, 0.00)
UTX	2.24583	4.31 ± 0.25	652	(0.01, 0.74)	(36.72, 0.00)	1.71142	4.01 ± 0.16	1433	(-4.69, 0.08)	(36.13, 0.01)	(40.82, 0.00)
V	2.01207	4.36 ± 0.22	890	(-0.04, 0.88)	(43.45, 0.00)	1.83352	4.17 ± 0.18	1230	(-1.23, 0.40)	(46.75, 0.00)	(47.98, 0.00)
VZ	1.93336	4.19 ± 0.2	982	(-0.00, 0.96)	(60.70, 0.00)	1.91867	4.28 ± 0.2	1030	(0.01, 0.74)	(67.94, 0.00)	(67.93, 0.00)
WBA	2.44431	4.32 ± 0.29	494	(-0.19, 0.71)	(18.64, 0.01)	1.87119	4.28 ± 0.19	1152	(-0.84, 0.46)	(42.17, 0.00)	(43.01, 0.00)
WMT	1.84977	4.02 ± 0.18	1107	(0.01, 0.57)	(83.56, 0.00)	1.92711	4.09 ± 0.19	966	(-0.00, 0.97)	(58.31, 0.00)	(58.31, 0.00)
XOM	1.84111	4.35 ± 0.19	1140	(0.02, 0.46)	(75.07, 0.00)	1.92338	4.41 ± 0.21	1013	(0.02, 0.60)	(73.27, 0.00)	(73.24, 0.00)

Table A.1: Power-law fit and comparison versus alternative distributions for 1-minute normalized returns. Fits are generated using the **powerLaw** python package [3] implementing the methodology from Clauset, Shalizi, and Newman [49]. Distributional comparisons are between power-law (PL), lognormal (LN), and exponential (Exp) distributions and give the log likelihood ratio and p-value of distribution 1 versus distribution 2.

symbol	Left					Right				
	$x_{\min}$	$\hat{\alpha}$	$N_{x \geq x_{\min}}$	$PL_{vs LN}(r,p)$	$PL_{vs LN}(r,p)$	$x_{\min}$	$\hat{\alpha}$	$N_{x \geq x_{\min}}$	$PL_{vs LN}(r,p)$	$PL_{vs LN}(r,p)$
AAPL	1.60332	6.54 ± 1.41	59	(-1.59, 0.25)	(-1.39, 0.07)	1.14605	4.47 ± 0.55	155	(-5.54, 0.04)	(-5.67, 0.00)
AXP	1.81017	7.78 ± 2.22	36	(-0.35, 0.57)	(-0.05, 0.61)	2.22676	22.59 ± 12.76	11	(-1.08, 0.02)	(0.18, 0.17)
BA	1.42132	5.61 ± 0.94	92	(-6.47, 0.02)	(-4.17, 0.00)	1.53980	5.97 ± 1.09	80	(-2.52, 0.15)	(-2.28, 0.02)
CAT	2.03912	12.69 ± 5.4	18	(-0.06, 0.80)	(-0.07, 0.73)	0.90513	3.46 ± 0.33	212	(-15.36, 0.00)	(-15.12, 0.00)
CSCO	1.33029	4.9 ± 0.72	113	(-6.02, 0.02)	(-4.90, 0.00)	1.92058	9.57 ± 3.36	25	(-0.08, 0.76)	(-0.09, 0.77)
CVX	1.23610	4.69 ± 0.63	131	(-6.29, 0.03)	(-5.75, 0.00)	1.87673	9.62 ± 2.81	36	(-0.05, 0.79)	(-0.03, 0.93)
DIS	1.56180	5.78 ± 1.1	73	(-3.34, 0.09)	(-2.60, 0.00)	1.37229	5.59 ± 0.95	89	(-2.29, 0.14)	(-2.29, 0.03)
DWDP	2.10710	15.08 ± 6.17	20	(-0.05, 0.80)	(-0.06, 0.75)	1.00473	3.77 ± 0.41	176	(-11.98, 0.00)	(-11.26, 0.00)
GS	2.13131	12.17 ± 4.57	23	(-0.05, 0.84)	(-0.06, 0.83)	2.12424	13.54 ± 5.36	21	(-0.04, 0.85)	(-0.05, 0.83)
HD	1.43388	5.75 ± 0.93	99	(-5.06, 0.04)	(-3.72, 0.00)	1.37370	5.46 ± 0.86	104	(-4.75, 0.05)	(-3.83, 0.00)
IBM	2.07822	13.1 ± 5.44	19	(-0.09, 0.75)	(-0.10, 0.65)	1.82570	8.42 ± 2.3	40	(-0.08, 0.44)	(-0.60, 0.21)
INTC	1.45977	5.88 ± 1.0	92	(-3.60, 0.09)	(-2.97, 0.00)	1.36065	5.99 ± 0.96	104	(-0.78, 0.37)	(-0.87, 0.46)
JNJ	1.49353	6.29 ± 1.18	77	(-2.14, 0.18)	(-1.91, 0.03)	1.92182	10.27 ± 3.43	28	(-0.63, 0.47)	(-0.42, 0.20)
JPM	1.01839	3.82 ± 0.4	189	(-11.83, 0.00)	(-11.25, 0.00)	1.49251	6.23 ± 1.15	79	(-2.46, 0.14)	(-2.11, 0.02)
KO	1.08081	4.14 ± 0.47	170	(-7.38, 0.01)	(-7.54, 0.00)	1.02813	3.81 ± 0.42	173	(-16.64, 0.00)	(-13.21, 0.00)
MCD	0.70476	2.91 ± 0.22	282	(-27.99, 0.00)	(-28.37, 0.00)	1.39241	6.16 ± 0.99	104	(-0.95, 0.33)	(-1.15, 0.31)
MMM	1.71645	7.65 ± 1.7	59	(-0.11, 0.72)	(-0.08, 0.91)	1.62344	6.58 ± 1.42	59	(-1.43, 0.24)	(-1.32, 0.07)
MRK	1.27869	4.65 ± 0.66	119	(-5.10, 0.03)	(-4.85, 0.00)	1.11850	4.32 ± 0.53	152	(-7.99, 0.01)	(-7.34, 0.00)
MSFT	1.61576	6.53 ± 1.42	58	(-0.89, 0.37)	(-1.00, 0.21)	1.60435	6.89 ± 1.39	69	(-2.07, 0.20)	(-1.69, 0.03)
NKE	1.54460	7.09 ± 1.36	77	(-2.49, 0.14)	(-1.88, 0.01)	1.49126	6.4 ± 1.18	81	(-1.99, 0.18)	(-1.85, 0.04)
PFE	1.16830	4.32 ± 0.55	140	(-6.72, 0.02)	(-6.44, 0.00)	1.25199	4.97 ± 0.7	123	(-4.11, 0.08)	(-4.17, 0.01)
PG	1.96896	9.79 ± 3.25	28	(-0.26, 0.61)	(-0.25, 0.45)	1.58885	8.01 ± 1.73	63	(-0.36, 0.57)	(-0.41, 0.57)
TRV	1.94235	10.98 ± 3.41	33	(-0.24, 0.65)	(-0.23, 0.51)	0.77822	3.13 ± 0.26	261	(-29.56, 0.00)	(-26.05, 0.00)
UNH	1.70556	7.94 ± 2.07	43	(-0.68, 0.41)	(-0.65, 0.20)	1.83577	8.35 ± 2.22	42	(-0.30, 0.61)	(-0.33, 0.55)
UTX	1.80224	8.77 ± 2.15	50	(-0.55, 0.48)	(-0.54, 0.30)	1.06932	3.91 ± 0.45	162	(-13.48, 0.00)	(-11.29, 0.00)
V	1.26542	4.62 ± 0.63	127	(-8.64, 0.01)	(-6.75, 0.00)	1.50021	6.41 ± 1.22	75	(-2.36, 0.14)	(-1.97, 0.02)
VZ	1.26146	4.78 ± 0.67	123	(-8.19, 0.01)	(-6.28, 0.00)	1.47432	5.72 ± 0.97	91	(-3.43, 0.08)	(-2.92, 0.00)
WBA	1.30732	4.9 ± 0.71	117	(-6.17, 0.02)	(-5.19, 0.00)	1.41348	5.61 ± 0.92	97	(-6.36, 0.03)	(-4.20, 0.00)
WMT	1.68984	6.76 ± 1.43	62	(-3.61, 0.09)	(-2.12, 0.00)	2.09149	10.64 ± 4.45	18	(-0.03, 0.85)	(-0.04, 0.89)
XOM	1.46883	6.26 ± 1.14	82	(-4.28, 0.06)	(-2.88, 0.00)	1.23108	4.5 ± 0.6	131	(-10.63, 0.00)	(-7.71, 0.00)

Table A.1: Power-law fit and comparison versus alternative distributions for 30-minute normalized returns. Fits are generated using the **powerLaw** python package [3] implementing the methodology from Clauset, Shalizi, and Newman [49]. Distributional comparisons are between power-law (PL), lognormal (LN), and exponential (Exp) distributions and give the log likelihood ratio and p-value of distribution 1 versus distribution 2.

symbol	Left					Right					
	x_{min}	$\hat{\alpha}$	$N_{Z\geq x_{min}}$	$PL_{vs_LN}(\tau, p)$	$LN_{vs_Exp}(\tau, p)$	x_{min}	$\hat{\alpha}$	$N_{Z\geq x_{min}}$	$PL_{vs_LN}(\tau, p)$	$PL_{vs_Exp}(\tau, p)$	$LN_{vs_Exp}(\tau, p)$
AAPL	1.36253	8.11 $\pm$ 2.08	45	(-2.61, 0.17)	(-1.30, 0.01)	1.42940	9.77 $\pm$ 2.43	50	(-0.44, 0.51)	(-0.46, 0.32)	(-0.02, 0.93)
AXP	1.06166	4.81 $\pm$ 0.8	87	(-7.15, 0.02)	(-4.91, 0.00)	1.50298	9.75 $\pm$ 3.03	32	(-1.79, 0.21)	(-0.75, 0.02)	(1.04, 0.36)
BA	1.50622	11.89 $\pm$ 3.56	36	(-0.37, 0.57)	(-0.32, 0.34)	1.09700	4.94 $\pm$ 0.84	85	(-9.36, 0.01)	(-5.34, 0.00)	(4.02, 0.12)
CAT	1.68772	19.91 $\pm$ 9.57	15	(0.00, 0.96)	(0.02, 0.88)	1.52221	10.04 $\pm$ 3.04	34	(-2.98, 0.12)	(-0.92, 0.00)	(2.05, 0.21)
CSCO	1.22964	5.98 $\pm$ 1.21	65	(-3.80, 0.08)	(-2.54, 0.00)	1.83018	42.02 $\pm$ 30.39	7	(-0.02, 0.00)	(0.04, 0.43)	(0.06, 0.24)
CVX	1.53056	10.18 $\pm$ 2.78	42	(-2.49, 0.17)	(-0.97, 0.01)	1.64080	29.27 $\pm$ 17.52	10	(-4.56, 0.00)	(0.04, 0.58)	(4.61, 0.00)
DIS	1.32430	7.65 $\pm$ 1.84	50	(-2.42, 0.19)	(-1.42, 0.01)	1.63509	18.59 $\pm$ 8.13	18	(-0.08, 0.76)	(-0.06, 0.63)	(0.01, 0.92)
DWDP	1.03084	4.8 $\pm$ 0.73	103	(-7.69, 0.01)	(-5.57, 0.00)	1.51042	11.23 $\pm$ 3.86	27	(-1.07, 0.35)	(-0.48, 0.08)	(0.59, 0.50)
GS	1.55537	12.24 $\pm$ 4.24	27	(-0.54, 0.44)	(-0.32, 0.17)	1.61073	17.72 $\pm$ 8.46	15	(-0.02, 0.88)	(-0.03, 0.85)	(-0.00, 0.43)
HD	1.24398	6.24 $\pm$ 1.31	61	(-5.66, 0.03)	(-2.76, 0.00)	1.46988	10.3 $\pm$ 3.0	37	(-1.31, 0.25)	(-0.67, 0.04)	(0.64, 0.44)
IBM	1.08769	4.77 $\pm$ 0.81	83	(-9.82, 0.01)	(-5.54, 0.00)	1.40536	7.7 $\pm$ 2.03	42	(-2.72, 0.15)	(-1.33, 0.01)	(1.38, 0.32)
INTC	1.11526	5.45 $\pm$ 0.96	82	(-5.04, 0.04)	(-3.57, 0.00)	0.83924	3.77 $\pm$ 0.48	129	(-12.74, 0.00)	(-10.24, 0.00)	(2.50, 0.27)
JNJ	0.85781	3.9 $\pm$ 0.5	130	(-9.76, 0.01)	(-8.69, 0.00)	0.75650	3.36 $\pm$ 0.38	146	(-16.21, 0.00)	(-13.83, 0.00)	(2.37, 0.33)
JPM	1.00412	4.5 $\pm$ 0.67	105	(-8.64, 0.01)	(-6.35, 0.00)	1.26470	7.08 $\pm$ 1.65	52	(-2.95, 0.12)	(-1.69, 0.00)	(1.26, 0.34)
KO	1.25850	6.59 $\pm$ 1.38	63	(-3.60, 0.09)	(-2.21, 0.00)	1.63947	17.85 $\pm$ 8.25	16	(-0.02, 0.90)	(-0.02, 0.89)	(-0.01, 0.83)
MCD	1.50660	10.36 $\pm$ 3.19	33	(-1.92, 0.20)	(-0.74, 0.01)	1.57011	12.84 $\pm$ 4.95	22	(-0.17, 0.68)	(-0.15, 0.51)	(0.02, 0.92)
MMM	1.20988	6.67 $\pm$ 1.36	67	(-1.81, 0.21)	(-1.59, 0.04)	1.75989	26.74 $\pm$ 13.48	14	(-0.23, 0.68)	(-0.07, 0.40)	(0.16, 0.74)
MRK	1.48930	10.37 $\pm$ 3.1	35	(-1.65, 0.26)	(-0.72, 0.03)	1.64555	19.15 $\pm$ 8.38	18	(-0.07, 0.80)	(-0.06, 0.69)	(0.01, 0.93)
MSFT	1.77383	22.95 $\pm$ 11.11	15	(-0.28, 0.59)	(-0.10, 0.30)	1.26035	6.31 $\pm$ 1.4	55	(-4.92, 0.05)	(-2.44, 0.00)	(2.47, 0.18)
NKE	1.50192	12.13 $\pm$ 3.54	38	(-0.55, 0.50)	(-0.39, 0.24)	1.16162	5.37 $\pm$ 0.99	75	(-8.65, 0.01)	(-4.35, 0.00)	(4.30, 0.07)
PFE	1.24111	5.92 $\pm$ 1.15	70	(-5.99, 0.03)	(-3.27, 0.00)	1.04786	5.24 $\pm$ 0.88	89	(-3.95, 0.05)	(-3.35, 0.00)	(0.60, 0.54)
PG	0.79842	3.38 $\pm$ 0.39	141	(-18.21, 0.00)	(-14.21, 0.00)	1.03458	5.28 $\pm$ 0.87	93	(-7.89, 0.01)	(-4.83, 0.00)	(3.07, 0.15)
TRV	1.48210	9.07 $\pm$ 2.57	38	(-2.82, 0.13)	(-1.08, 0.00)	1.51284	10.13 $\pm$ 3.32	29	(-0.88, 0.40)	(-0.51, 0.12)	(0.38, 0.60)
UNH	1.67566	19.87 $\pm$ 8.72	18	(-0.76, 0.43)	(-0.19, 0.12)	1.64754	16.03 $\pm$ 6.94	18	(-0.01, 0.93)	(-0.01, 0.98)	(0.00, 0.99)
UTX	1.38374	8.24 $\pm$ 2.05	48	(-1.90, 0.20)	(-1.15, 0.02)	1.31571	7.4 $\pm$ 1.81	48	(-4.39, 0.06)	(-1.82, 0.00)	(2.58, 0.16)
V	1.76569	21.79 $\pm$ 9.61	18	(-0.34, 0.59)	(-0.12, 0.30)	1.31523	7.52 $\pm$ 1.76	53	(-2.39, 0.17)	(-1.47, 0.01)	(0.93, 0.43)
VZ	0.96492	4.55 $\pm$ 0.67	107	(-7.27, 0.02)	(-5.85, 0.00)	0.81687	3.4 $\pm$ 0.41	129	(-17.28, 0.00)	(-13.18, 0.00)	(4.11, 0.14)
WBA	1.83561	51.4 $\pm$ 37.34	7	(-0.72, 0.00)	(0.02, 0.52)	1.16998	5.78 $\pm$ 1.12	70	(-6.54, 0.02)	(-3.46, 0.00)	(3.08, 0.14)
WMT	1.82187	23.89 $\pm$ 11.99	14	(-0.71, 0.47)	(-0.13, 0.14)	1.43363	9.1 $\pm$ 2.76	33	(-1.91, 0.18)	(-0.84, 0.01)	(1.07, 0.33)
XOM	1.69901	21.21 $\pm$ 10.99	13	(-0.01, 0.91)	(-0.02, 0.88)	1.31011	7.12 $\pm$ 1.66	52	(-3.06, 0.14)	(-1.73, 0.01)	(1.34, 0.36)

Table A.1: Power-law fit and comparison versus alternative distributions for 60-minute normalized returns. Fits are generated using the `powerLaw` python package [3] implementing the methodology from Clauset, Shalizi, and Newman [49]. Distributional comparisons are between power-law (PL), lognormal (LN), and exponential (Exp) distributions and give the log likelihood ratio and p-value of distribution 1 versus distribution 2.

symbol	Left					Right					
	x_{min}	$\hat{\alpha}$	$N_{x \geq x_{min}}$	$PL_{us_LN}(\tau, p)$	$LN_{us_exp}(\tau, p)$	x_{min}	$\hat{\alpha}$	$N_{x \geq x_{min}}$	$PL_{us_LN}(\tau, p)$	$PL_{us_exp}(\tau, p)$	$LN_{us_exp}(\tau, p)$
AAPL	0.48663	2.97 ± 0.0	892893	(54.32, 0.00)	(418634.83, 0.00)	0.48689	2.96 ± 0.0	888023	(53.32, 0.00)	(413451.81, 0.00)	(413398.48, 0.00)
AXP	2.81148	3.0 ± 0.03	16645	(-0.12, 0.80)	(3023.77, 0.00)	2.86540	2.99 ± 0.03	15882	(-0.25, 0.72)	(2875.79, 0.00)	(2876.04, 0.00)
BA	1.42377	3.45 ± 0.01	120035	(6.09, 0.00)	(31535.98, 0.00)	1.37179	3.43 ± 0.01	130648	(5.17, 0.00)	(32854.92, 0.00)	(32849.75, 0.00)
CAT	1.29117	3.27 ± 0.01	117109	(2.18, 0.00)	(26343.61, 0.00)	1.29073	3.26 ± 0.01	118071	(2.41, 0.00)	(26326.55, 0.00)	(26324.14, 0.00)
CSCO	3.16336	2.4 ± 0.02	18034	(-2.32, 0.15)	(5375.27, 0.00)	3.44199	2.41 ± 0.02	16190	(-2.17, 0.16)	(4694.83, 0.00)	(4697.00, 0.00)
CVX	0.65199	3.0 ± 0.01	279368	(2.85, 0.00)	(74929.55, 0.00)	0.65205	3.01 ± 0.01	280391	(3.05, 0.00)	(75379.54, 0.00)	(75376.48, 0.00)
DIS	0.66238	3.06 ± 0.01	396902	(6.77, 0.00)	(98488.83, 0.00)	2.61596	2.76 ± 0.02	21619	(0.87, 0.00)	(6907.02, 0.00)	(6906.15, 0.00)
DWDP	3.45169	2.29 ± 0.03	9367	(-1.87, 0.14)	(2561.64, 0.00)	3.10428	2.32 ± 0.02	11209	(0.02, 0.82)	(3401.32, 0.00)	(3401.30, 0.00)
GS	2.23091	3.64 ± 0.02	51432	(0.84, 0.30)	(8181.09, 0.00)	2.21424	3.65 ± 0.02	52855	(0.98, 0.02)	(8380.71, 0.00)	(8379.73, 0.00)
HD	0.99075	3.08 ± 0.01	140890	(5.82, 0.00)	(43137.05, 0.00)	1.00369	3.08 ± 0.01	137867	(4.69, 0.00)	(41969.21, 0.00)	(41964.52, 0.00)
IBM	0.84079	3.05 ± 0.01	143997	(7.19, 0.00)	(50157.68, 0.00)	0.90078	3.05 ± 0.01	128544	(6.84, 0.00)	(45158.97, 0.00)	(45152.12, 0.00)
INTC	3.52231	2.4 ± 0.02	17037	(-1.20, 0.30)	(5191.50, 0.00)	3.52391	2.4 ± 0.02	17088	(-1.55, 0.24)	(5181.54, 0.00)	(5183.10, 0.00)
JNJ	0.73801	2.91 ± 0.01	215924	(6.79, 0.00)	(75443.35, 0.00)	0.74360	2.9 ± 0.01	214357	(6.50, 0.00)	(74331.52, 0.00)	(74325.02, 0.00)
JPM	3.87086	2.27 ± 0.03	9855	(-2.81, 0.08)	(2899.43, 0.00)	4.60768	2.23 ± 0.03	7680	(-16.33, 0.00)	(1831.92, 0.00)	(1848.25, 0.00)
KO	2.94795	2.53 ± 0.03	14054	(0.06, 0.76)	(4036.21, 0.00)	3.27669	2.53 ± 0.03	11995	(-0.11, 0.77)	(3313.09, 0.00)	(3313.20, 0.00)
MCD	1.25527	3.16 ± 0.01	94684	(1.96, 0.00)	(21706.05, 0.00)	1.25782	3.16 ± 0.01	94970	(0.66, 0.51)	(21674.51, 0.00)	(21673.85, 0.00)
MMM	1.36718	3.33 ± 0.02	59790	(4.03, 0.00)	(16223.26, 0.00)	1.36824	3.32 ± 0.02	60440	(2.60, 0.00)	(16059.77, 0.00)	(16057.17, 0.00)
MRK	4.27616	2.55 ± 0.03	10170	(-0.70, 0.42)	(2448.00, 0.00)	4.28002	2.55 ± 0.03	10206	(-0.66, 0.43)	(2456.39, 0.00)	(2457.05, 0.00)
MSFT	2.95416	2.35 ± 0.02	29798	(0.11, 0.51)	(11107.68, 0.00)	2.77176	2.35 ± 0.01	32640	(0.05, 0.85)	(12269.94, 0.00)	(12269.89, 0.00)
NKE	3.24047	2.82 ± 0.03	15430	(-6.56, 0.06)	(2747.85, 0.00)	3.25302	2.84 ± 0.03	15706	(-5.66, 0.08)	(2789.70, 0.00)	(2795.36, 0.00)
PFE	2.70753	2.42 ± 0.02	19465	(-0.03, 0.90)	(6308.94, 0.00)	2.70960	2.42 ± 0.02	19678	(0.08, 0.51)	(6443.62, 0.00)	(6443.54, 0.00)
PG	2.88767	2.39 ± 0.02	12840	(0.03, 0.83)	(4002.67, 0.00)	2.88952	2.4 ± 0.02	13039	(0.10, 0.37)	(4092.62, 0.00)	(4092.51, 0.00)
TRV	2.12857	3.51 ± 0.03	24151	(0.56, 0.00)	(3960.21, 0.00)	2.28194	3.52 ± 0.03	20530	(0.27, 0.13)	(3569.54, 0.00)	(3569.28, 0.00)
UNH	1.48880	3.3 ± 0.02	71448	(4.35, 0.00)	(19784.26, 0.00)	1.47941	3.3 ± 0.02	72577	(4.15, 0.00)	(20015.24, 0.00)	(20011.09, 0.00)
UTX	1.29066	3.16 ± 0.01	96092	(2.65, 0.00)	(22467.76, 0.00)	1.20950	3.15 ± 0.01	107971	(0.60, 0.53)	(24673.36, 0.00)	(24672.76, 0.00)
V	4.44314	2.47 ± 0.03	7015	(-5.71, 0.01)	(2182.44, 0.00)	4.29369	2.49 ± 0.03	7518	(-7.20, 0.00)	(2387.76, 0.00)	(2394.96, 0.00)
VZ	2.76873	2.39 ± 0.02	17154	(-1.00, 0.33)	(5139.25, 0.00)	2.77304	2.41 ± 0.02	17385	(-0.26, 0.62)	(5259.22, 0.00)	(5259.48, 0.00)
WBA	4.79099	2.84 ± 0.04	7868	(-3.18, 0.14)	(1296.16, 0.00)	5.78105	2.89 ± 0.05	5727	(-0.61, 0.50)	(933.20, 0.00)	(933.81, 0.00)
WMT	3.13830	2.63 ± 0.03	14543	(0.14, 0.11)	(3979.31, 0.00)	2.93557	2.62 ± 0.02	16215	(0.17, 0.22)	(4415.83, 0.00)	(4415.66, 0.00)
XOM	3.13256	2.32 ± 0.02	12013	(-1.92, 0.15)	(3653.52, 0.00)	2.85186	2.35 ± 0.02	14099	(0.08, 0.59)	(4655.70, 0.00)	(4655.62, 0.00)

Table A.1: Power-law fit and comparison versus alternative distributions for trade-level normalized returns. Fits are generated using the **powerLaw** python package [3] implementing the methodology from Clauset, Shalizi, and Newman [49]. Distributional comparisons are between power-law (PL), lognormal (LN), and exponential (Exp) distributions and give the log likelihood ratio and p-value of distribution 1 versus distribution 2.

symbol	Left					Right						
	$x_{\min}$	$\hat{\alpha}$	$N_{x \geq x_{\min}}$	$PL_{us_LN}(r,p)$	$PL_{us_Exp}(r,p)$	$LN_{us_exp}(r,p)$	$x_{\min}$	$\hat{\alpha}$	$N_{x \geq x_{\min}}$	$PL_{us_LN}(r,p)$	$PL_{us_exp}(r,p)$	$LN_{us_exp}(r,p)$
AAPL	1.56337	4.72 ± 0.07	9690	(0.67, 0.00)	(1466.78, 0.00)	(1466.11, 0.00)	1.58032	4.71 ± 0.07	9773	(0.22, 0.48)	(1422.93, 0.00)	(1422.71, 0.00)
AXP	2.08648	5.08 ± 0.29	751	(-7.81, 0.02)	(-6.46, 0.20)	(1.35, 0.44)	2.37881	5.22 ± 0.38	467	(-0.30, 0.62)	(7.75, 0.10)	(8.05, 0.05)
BA	2.23040	5.39 ± 0.29	904	(-0.58, 0.40)	(131.37, 0.00)	(131.94, 0.00)	2.08552	5.25 ± 0.24	1183	(-0.58, 0.41)	(145.20, 0.00)	(145.78, 0.00)
CAT	2.78853	6.08 ± 0.54	340	(0.00, 0.89)	(10.93, 0.12)	(10.93, 0.12)	2.78505	5.93 ± 0.49	395	(-0.64, 0.35)	(12.33, 0.08)	(12.98, 0.09)
CSCO	1.74267	4.84 ± 0.12	4049	(-0.32, 0.19)	(302.44, 0.00)	(302.77, 0.00)	1.99504	5.02 ± 0.15	2809	(-1.77, 0.03)	(258.14, 0.00)	(259.91, 0.00)
CVX	1.99763	5.19 ± 0.23	1234	(-0.63, 0.20)	(82.11, 0.00)	(82.73, 0.00)	2.40072	5.33 ± 0.33	649	(-0.81, 0.04)	(53.94, 0.00)	(54.75, 0.00)
DIS	2.02510	4.7 ± 0.19	1430	(0.01, 0.71)	(79.65, 0.00)	(79.63, 0.00)	2.18355	5.06 ± 0.24	1124	(-1.47, 0.09)	(81.30, 0.00)	(82.77, 0.00)
DWDP	2.14006	5.03 ± 0.22	1294	(-0.75, 0.27)	(137.71, 0.00)	(138.46, 0.00)	1.98973	4.82 ± 0.18	1702	(-0.89, 0.24)	(154.90, 0.00)	(155.79, 0.00)
GS	2.06567	5.32 ± 0.27	959	(-6.51, 0.07)	(-1.61, 0.83)	(4.90, 0.24)	2.88779	6.19 ± 0.62	266	(-0.50, 0.60)	(1.37, 0.69)	(1.87, 0.46)
HD	2.35235	4.98 ± 0.29	710	(-1.17, 0.12)	(55.63, 0.00)	(56.80, 0.00)	1.96188	4.76 ± 0.2	1330	(0.01, 0.75)	(78.33, 0.00)	(78.32, 0.00)
IBM	2.07606	4.77 ± 0.25	905	(-0.98, 0.15)	(73.68, 0.00)	(74.66, 0.00)	1.96683	4.77 ± 0.2	1309	(-0.13, 0.20)	(83.49, 0.00)	(83.62, 0.00)
INTC	1.85667	5.0 ± 0.13	3771	(-1.59, 0.08)	(355.09, 0.00)	(356.68, 0.00)	1.71945	4.73 ± 0.1	5197	(0.05, 0.62)	(364.12, 0.00)	(364.06, 0.00)
JNJ	1.71589	4.24 ± 0.14	2138	(-0.84, 0.16)	(208.60, 0.00)	(209.43, 0.00)	1.81557	4.26 ± 0.15	1813	(0.08, 0.28)	(196.26, 0.00)	(196.17, 0.00)
JPM	1.97133	5.2 ± 0.17	2360	(-1.06, 0.17)	(255.99, 0.00)	(257.05, 0.00)	2.00060	5.09 ± 0.16	2466	(0.15, 0.01)	(252.22, 0.00)	(252.06, 0.00)
KO	2.10864	4.79 ± 0.19	1479	(0.01, 0.84)	(114.15, 0.00)	(114.14, 0.00)	2.11062	4.74 ± 0.19	1470	(-0.88, 0.04)	(92.44, 0.00)	(93.32, 0.00)
MCD	2.23412	4.66 ± 0.28	657	(-0.06, 0.85)	(25.89, 0.01)	(25.96, 0.01)	1.83264	4.59 ± 0.2	1192	(-0.03, 0.89)	(52.49, 0.00)	(52.52, 0.00)
MMM	2.38915	5.07 ± 0.41	374	(0.01, 0.27)	(21.32, 0.00)	(21.31, 0.00)	2.29936	5.43 ± 0.41	449	(-0.52, 0.05)	(21.29, 0.01)	(21.82, 0.01)
MRK	1.95670	5.0 ± 0.18	1896	(0.02, 0.69)	(84.09, 0.00)	(84.07, 0.00)	2.36741	4.94 ± 0.24	994	(-0.65, 0.26)	(47.79, 0.00)	(48.44, 0.00)
MSFT	1.60716	4.83 ± 0.08	9847	(-0.77, 0.39)	(905.08, 0.00)	(905.85, 0.00)	1.84703	5.01 ± 0.1	6590	(0.43, 0.00)	(687.64, 0.00)	(687.21, 0.00)
NKE	2.48528	5.64 ± 0.39	538	(-0.85, 0.07)	(25.78, 0.02)	(26.63, 0.02)	2.14600	4.91 ± 0.24	1049	(-1.00, 0.48)	(24.07, 0.05)	(25.07, 0.02)
PFE	1.85767	4.87 ± 0.14	2944	(-0.33, 0.11)	(342.31, 0.00)	(342.63, 0.00)	1.87389	4.76 ± 0.14	2934	(-0.11, 0.39)	(311.52, 0.00)	(311.63, 0.00)
PG	1.87668	4.51 ± 0.16	1873	(-0.10, 0.39)	(148.45, 0.00)	(148.54, 0.00)	1.84950	4.39 ± 0.15	1895	(0.03, 0.41)	(148.84, 0.00)	(148.81, 0.00)
TRV	3.04105	6.47 ± 1.09	96	(-0.02, 0.88)	(1.06, 0.43)	(1.08, 0.37)	1.98987	5.1 ± 0.38	445	(-0.37, 0.66)	(9.65, 0.15)	(10.02, 0.09)
UNH	2.22528	4.81 ± 0.28	703	(0.03, 0.46)	(58.15, 0.00)	(58.12, 0.00)	2.37851	5.03 ± 0.32	621	(-1.05, 0.16)	(48.81, 0.01)	(49.86, 0.01)
UTX	2.21508	4.89 ± 0.27	778	(-0.00, 0.97)	(27.66, 0.00)	(27.66, 0.00)	2.06537	4.68 ± 0.24	941	(-1.38, 0.38)	(21.28, 0.04)	(22.66, 0.01)
V	2.07819	4.93 ± 0.2	1469	(-0.39, 0.05)	(110.13, 0.00)	(110.52, 0.00)	2.24367	4.73 ± 0.21	1195	(-1.18, 0.05)	(88.38, 0.00)	(89.56, 0.00)
VZ	2.04719	4.76 ± 0.17	1869	(0.02, 0.50)	(102.20, 0.00)	(102.17, 0.00)	2.19041	4.73 ± 0.19	1519	(0.02, 0.61)	(81.57, 0.00)	(81.55, 0.00)
WBA	2.41575	4.87 ± 0.31	608	(-0.05, 0.86)	(19.23, 0.02)	(19.28, 0.01)	2.59406	4.86 ± 0.35	468	(-0.02, 0.92)	(16.72, 0.04)	(16.74, 0.03)
WMT	1.78795	4.41 ± 0.15	1992	(0.00, 1.00)	(101.93, 0.00)	(101.93, 0.00)	2.15617	4.66 ± 0.21	1131	(0.04, 0.14)	(76.31, 0.00)	(76.27, 0.00)
XOM	2.04257	5.32 ± 0.2	1762	(-0.71, 0.19)	(114.56, 0.00)	(115.28, 0.00)	1.91942	5.24 ± 0.17	2340	(-0.20, 0.22)	(140.58, 0.00)	(140.78, 0.00)

Table A.1: Power-law fit and comparison versus alternative distributions for 100-trade normalized returns. Fits are generated using the `powerLaw` python package [3] implementing the methodology from Clauset, Shalizi, and Newman [49]. Distributional comparisons are between power-law (PL), lognormal (LN), and exponential (Exp) distributions and give the log likelihood ratio and p-value of distribution 1 versus distribution 2.

symbol	Left					Right				
	$x_{\min}$	$\hat{\alpha}$	$N_{x \geq x_{\min}}$	$PL_{vs LN}(\tau, p)$	$LN_{vs Exp}(\tau, p)$	$PL_{vs Exp}(\tau, p)$	$LN_{vs LN}(\tau, p)$	$PL_{vs Exp}(\tau, p)$	$LN_{vs Exp}(\tau, p)$	$LN_{vs Exp}(\tau, p)$
AAPL	2.10193	6.24 ± 0.5	428	(0.01, 0.58)	(15.98, 0.05)	(15.97, 0.05)	(-1.11, 0.02)	(29.92, 0.00)	(31.03, 0.00)	(31.03, 0.00)
AXP	1.75790	6.25 ± 0.89	133	(-1.76, 0.25)	(-1.87, 0.19)	(-0.11, 0.50)	(-25.31, 0.00)	(-27.31, 0.00)	(-2.00, 0.08)	(-2.00, 0.08)
BA	2.14195	6.94 ± 1.11	110	(-1.61, 0.25)	(-1.65, 0.12)	(-0.04, 0.91)	(-1.75, 0.23)	(-1.84, 0.28)	(-0.09, 0.74)	(-0.09, 0.74)
CAT	1.82260	5.76 ± 0.69	182	(-4.64, 0.05)	(-4.52, 0.00)	(0.12, 0.89)	(-1.11, 0.28)	(-1.31, 0.22)	(-0.21, 0.02)	(-0.21, 0.02)
CSCO	2.25304	6.81 ± 0.86	175	(-1.84, 0.23)	(-1.99, 0.18)	(-0.15, 0.20)	(-0.34, 0.59)	(1.67, 0.49)	(2.01, 0.26)	(2.01, 0.26)
CVX	1.86876	5.99 ± 0.72	182	(-1.06, 0.41)	(-0.64, 0.78)	(0.42, 0.67)	(-0.10, 0.80)	(1.07, 0.60)	(1.17, 0.47)	(1.17, 0.47)
DIS	2.08959	7.12 ± 1.05	130	(-0.41, 0.55)	(-0.22, 0.86)	(0.19, 0.77)	(-1.18, 0.31)	(-1.27, 0.22)	(-0.09, 0.53)	(-0.09, 0.53)
DWDP	2.47595	6.26 ± 1.11	87	(0.00, 0.85)	(2.26, 0.34)	(2.25, 0.34)	(-0.02, 0.89)	(0.90, 0.53)	(0.92, 0.47)	(0.92, 0.47)
GS	2.26424	8.12 ± 1.73	65	(-0.25, 0.66)	(-0.23, 0.78)	(0.02, 0.93)	(-0.03, 0.85)	(0.06, 0.90)	(0.09, 0.79)	(0.09, 0.79)
HD	2.22989	7.26 ± 1.29	91	(-0.03, 0.89)	(0.75, 0.58)	(0.78, 0.51)	(-25.40, 0.00)	(-27.43, 0.00)	(-2.02, 0.01)	(-2.02, 0.01)
IBM	1.81528	5.67 ± 0.69	175	(-0.94, 0.38)	(-0.47, 0.82)	(0.48, 0.62)	(-13.06, 0.00)	(-13.73, 0.00)	(-0.68, 0.11)	(-0.68, 0.11)
INTC	1.65071	5.29 ± 0.32	672	(-15.21, 0.00)	(-14.97, 0.00)	(0.24, 0.79)	(-0.58, 0.47)	(-0.68, 0.51)	(-0.10, 0.68)	(-0.10, 0.68)
JNJ	1.70576	5.12 ± 0.49	268	(-1.56, 0.45)	(1.24, 0.81)	(2.80, 0.39)	(-0.03, 0.44)	(3.26, 0.14)	(3.29, 0.14)	(3.29, 0.14)
JPM	2.24463	6.97 ± 0.92	161	(0.00, 0.89)	(3.12, 0.37)	(3.12, 0.37)	(-0.05, 0.88)	(2.89, 0.42)	(2.94, 0.37)	(2.94, 0.37)
KO	1.78602	5.02 ± 0.47	277	(-2.70, 0.16)	(-2.42, 0.43)	(0.29, 0.80)	(-2.34, 0.15)	(-1.44, 0.65)	(0.91, 0.58)	(0.91, 0.58)
MCD	1.41478	4.58 ± 0.42	276	(-7.20, 0.01)	(-7.81, 0.00)	(-0.62, 0.15)	(-9.96, 0.00)	(-9.50, 0.00)	(0.46, 0.72)	(0.46, 0.72)
MMM	1.87122	6.78 ± 1.19	90	(-0.14, 0.71)	(0.29, 0.81)	(0.43, 0.58)	(-0.00, 0.98)	(0.33, 0.63)	(0.33, 0.62)	(0.33, 0.62)
MRK	1.90919	5.97 ± 0.63	238	(-1.11, 0.38)	(-0.38, 0.88)	(0.72, 0.56)	(-0.10, 0.75)	(0.41, 0.74)	(0.51, 0.57)	(0.51, 0.57)
MSFT	2.26741	6.58 ± 0.62	313	(-0.69, 0.17)	(17.13, 0.06)	(17.82, 0.06)	(-2.35, 0.37)	(2.91, 0.73)	(5.26, 0.36)	(5.26, 0.36)
NKE	1.53119	4.72 ± 0.44	279	(-6.86, 0.02)	(-7.23, 0.00)	(-0.37, 0.36)	(-4.75, 0.06)	(-4.97, 0.07)	(-0.22, 0.54)	(-0.22, 0.54)
PFE	2.54405	7.18 ± 1.22	99	(-1.67, 0.11)	(2.24, 0.20)	(3.91, 0.14)	(-0.11, 0.73)	(2.06, 0.30)	(2.17, 0.20)	(2.17, 0.20)
PG	2.10147	5.88 ± 0.78	152	(-1.10, 0.34)	(-1.03, 0.55)	(0.07, 0.90)	(-2.70, 0.14)	(-2.90, 0.19)	(-0.20, 0.61)	(-0.20, 0.61)
TRV	1.27338	4.57 ± 0.54	168	(-11.51, 0.00)	(-8.99, 0.00)	(2.53, 0.20)	(-0.18, 0.64)	(-0.16, 0.46)	(0.02, 0.90)	(0.02, 0.90)
UNH	1.88988	6.38 ± 0.87	147	(-1.19, 0.42)	(-1.16, 0.56)	(0.03, 0.95)	(0.00, 0.95)	(0.23, 0.62)	(0.23, 0.61)	(0.23, 0.61)
UTX	2.05590	6.2 ± 0.95	115	(-2.79, 0.15)	(-2.60, 0.03)	(0.18, 0.80)	(-2.85, 0.17)	(-2.96, 0.14)	(-0.11, 0.53)	(-0.11, 0.53)
V	1.72031	5.41 ± 0.49	313	(-5.61, 0.05)	(-5.84, 0.03)	(-0.22, 0.48)	(-0.95, 0.44)	(-0.57, 0.79)	(0.38, 0.70)	(0.38, 0.70)
VZ	1.67830	5.31 ± 0.42	414	(-2.19, 0.24)	(0.07, 0.99)	(2.26, 0.36)	(-0.00, 0.88)	(0.15, 0.69)	(0.15, 0.68)	(0.15, 0.68)
WBA	1.28629	4.16 ± 0.3	432	(-9.72, 0.01)	(-10.57, 0.01)	(-0.84, 0.18)	(-0.17, 0.71)	(0.92, 0.62)	(1.09, 0.44)	(1.09, 0.44)
WMT	1.54120	4.34 ± 0.35	354	(-6.54, 0.02)	(-6.95, 0.03)	(-0.41, 0.49)	(-3.79, 0.06)	(-4.11, 0.03)	(-0.33, 0.19)	(-0.33, 0.19)
XOM	1.75795	5.67 ± 0.51	321	(-2.50, 0.21)	(-1.79, 0.56)	(0.71, 0.53)	(-0.07, 0.79)	(-0.07, 0.80)	(-0.01, 0.85)	(-0.01, 0.85)

Table A.1: Power-law fit and comparison versus alternative distributions for 1000-trade normalized returns. Fits are generated using the `powerLaw` python package [3] implementing the methodology from Clauset, Shalizi, and Newman [49]. Distributional comparisons are between power-law (PL), lognormal (LN), and exponential (Exp) distributions and give the log likelihood ratio and p-value of distribution 1 versus distribution 2.

APPENDIX B

SUPPLEMENTARY MATERIAL FOR CHAPTER 3

B.1 USING *MARKETSIM* TO REPLICATE WWW

In order to validate our understanding and implementation of the *MarketSim* codebase, we attempt to use it to replicate the results from Wah, Wright, and Wellman (WWW) [191]. WWW cite the *MarketSim* GitHub repo¹ as the source code for the simulator used in their study. Our expectation is therefore that *MarketSim*, as published, should be able to replicate the WWW results. Success in doing so would help establish confidence in our implementation of the codebase for the purpose of replicating the earlier study by Wah and Wellman (WW) [189].

As noted in our main paper, WWW is detailed with similar assumptions and logic as the *CDA* configurations of the WW model. In the *CDA* configuration, the

¹The original *MarketSim* repo cited by WWW is here. We fork this repo to configure the replication experiments for WW and WWW here.

main differences between WW and WWW are the environment parameters used in their respective experiments and the introduction of market maker agents in WWW. WWW features experiments with and without market maker agents in five different configuration environments. We limit our replication exercise to the environments **A** and **B** with 66 background traders (ZI) and without a market maker:

$$\mathbf{A}: \lambda = 0.0005, \sigma_{\text{shock}}^2 = 1 \times 10^6, \sigma_{PV}^2 = 5 \times 10^6$$

$$\mathbf{B}: \lambda = 0.005, \sigma_{\text{shock}}^2 = 1 \times 10^6, \sigma_{PV}^2 = 5 \times 10^6$$

These settings are respectively used for each $\mathbf{A}\{T\}$ and $\mathbf{B}\{T\}$, where $T \in \{1, 4, 12, 24\}$ and the simulation length is $T \times 1000$.

WWW uses a similar experimental design to WW. They use the ‘empirical game-theoretic analysis’ (EGTA) approach to determine the surplus-maximizing equilibria ZI strategy mixtures. They then sample M profiles from the equilibrium mixture, running N simulations per mixture. In WWW, they specify the following:

For each equilibrium, we estimated background-trader surplus by sampling 2,500 profiles according to the equilibrium mixture, running 25–100 simulations per sampled profile (at least 62,500 simulations in total) and then recording the aggregate surplus. [191, p. 625]

They do not specify the specific number of runs used for each experiment, complicating the replication process. The mixtures of background trader profiles for each $\mathbf{A}\{T\}$ and $\mathbf{B}\{T\}$ are based on Tables 1, 2, and 9 in Wah, Wright, and Wellman [191], which, respectively, give the specific ZI strategies, the surplus-maximizing equilibria, and profile mixtures for the equilibria.

We sample 25,000 profiles and run 25 simulations for each experiment. We then use the bootstrap quantitative alignment methodology introduced in our main paper to test our results for quantitative alignment against the mean surplus values reported by WWW. The WWW means fall within the 95% confidence intervals (CIs) of our bootstrap sample means for six out of the eight environments tested. The reported mean for Environment **B24** fell within the 99% CI but not the 95% CI, and the reported mean for Environment **A24** falls outside the range of our bootstrap samples. By our replication criteria, we do not reject quantitative alignment between our *MarketSim* results and WWW for six out of eight environments at the 95% confidence level and seven out of eight environments at the 99% confidence level.

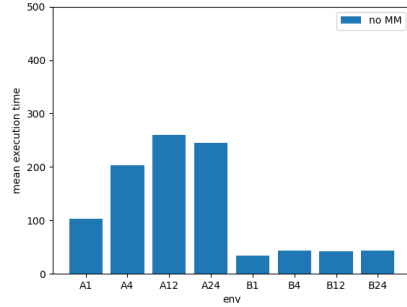
It is not clear what is driving the difference observed in Environment **A24**. As noted above, WWW state that each experiment was run for 25–100 simulations per sampled profile mixture, but they do not specify which experiment was run for what number of simulations per mixture. It is possible WWW ran **A24** for more than 25 simulations, that the original result is from the tail of possible sample means that can be produced by *MarketSim* for the **A24** experiment, or that there is some other small difference between how we are running the experiment and what was done in the original study.

env	WWW	<i>MarketSim</i> Bootstrap ZI Surplus			
	mean	mean	SE	95% CI diff.	99% CI diff.
A1	4439	4444.38	8.05	(-10.97, 20.87)	(-14.45, 25.33)
A4	16578	16572.59	15.61	(-34.96, 26.31)	(-45.73, 35.12)
A12	33741	33699.51	23.06	(-88.31, 0.22)	(-103.14, 15.86)
A24	42413	42516.26	28.99	(45.27, 160.30)	(30.83, 173.83)
B1	29150	29132.54	22.81	(-61.62, 25.37)	(-73.20, 40.95)
B4	40392	40352.06	30.62	(-102.48, 20.33)	(-120.91, 41.94)
B12	40102	40075.14	30.65	(-87.57, 29.82)	(-103.35, 52.00)
B24	40170	40234.19	29.84	(4.18, 124.34)	(-8.61, 139.32)

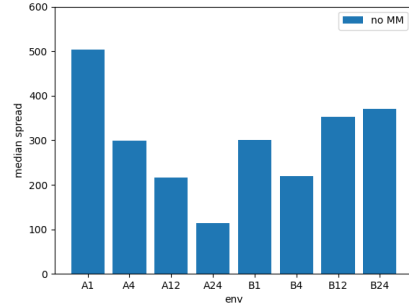
Table B.1: Test of quantitative alignment of *MarketSim* against the results reported by WWW. Environment parameters are as specified by WWW. Each ‘WWW’ value is the mean ZI surplus value reported by Wah, Wright, and Wellman [191] from 2,500 mixtures and 25–100 runs per mixture, for the given experimental settings. We ran *MarketSim* for 25,000 mixtures, 25 runs per mixture. From these results, we calculate 1,000 bootstrap samples of 2,500 mixtures, 25 runs per mixture, for each experiment. We report the mean and standard error of the bootstrap sample means, along with the 95% confidence intervals of the difference between the bootstrap sample means and the WWW mean. Bootstrap confidence intervals that contain the WWW mean are highlighted in **bold**.

We test relational equivalence by plotting the execution time and median spread metrics in Fig. B.1 for comparison against the corresponding figures in WWW (see Fig. 5 in Wah, Wright, and Wellman [191], p. 633). Note that we did not run the experiments with a market maker from WWW, so our plots only correspond to the ‘no MM’ results (shown in blue in each plot). We see close alignment of the results, with similar quantitative values and relative differences between environments. The most notable difference is that **A12** has a slightly higher execution time than **A24** in our run of *MarketSim*, while the opposite is true in WWW. The mean execution times are similar between these two environments in both sets of results. We are not able to otherwise visually distinguish the two sets of results. For now, we take the similarity of results (quantitative alignment and relational equivalence in most settings) as positive sign we have implemented and are running *MarketSim* correctly.

The fact *MarketSim* can produce results that closely align to the *WWW* results but not the *WW* results (see our results in the main paper) raises questions about the logic used in the published results. Wah’s dissertation [188] describes common modeling assumptions for *WW* and *WWW* and does not appear to differentiate these models aside from their parameters and market configurations. The *CDA* market mechanism and *ZI* trader strategy in particular are given common descriptions for the two models. We would therefore expect *MarketSim* as published for the *WWW* study to reproduce the results from the *CDA* configurations of *WW*. As we are not able to reproduce the *WW* results using *MarketSim* here, this suggests logic differences beyond what was specified in the original papers, perhaps in the *ZI* logic or in the logic used to calculate the output metrics at the end of the simulation.



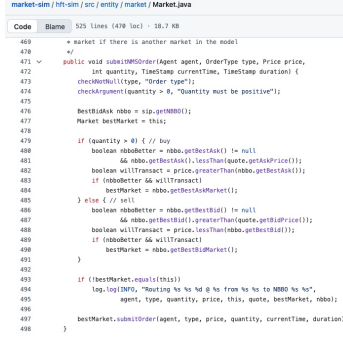
(a) *MarketSim* execution time results.



(b) *MarketSim* spread results

Figure B.1: Execution time and spread results from our *MarketSim* runs of *WWW* experiments for Environments **A** and **B**, $N = 66$, and no market maker. These results should be compared to the corresponding original figures from *WWW* [191, Fig. 5, p. 633]. Our results show the mean value produced by *MarketSim* for 25,000 mixtures, 25 runs per mixture.

B.2 TESTING MODEL IMPLEMENTATION WITH ORDER ROUTING BUG



```

market-sim / nb-sim / src / entity / market / Market.java
Code 325 lines (478 loc) - 16.7 KB
469 // market if there is another market in the model
470 //
471 public void submitOrder(Agent agent, OrderType type, Price price,
472     int quantity, Timestamp currentTime, Timestamp duration) {
473     checkNotNull(type, "Order type");
474     checkArgument(quantity > 0, "Quantity must be positive");
475
476     BestAsk nbbo = sip.getNBBO();
477     Market bestMarket = this;
478
479     if (quantity > 0) { // buy
480         boolean nbobetter = nbbo.getBestAsk() != null
481             && nbbo.getBestAsk().lessThanQuote.getAskPrice();
482         boolean willTransact = price.greaterThan(nbbo.getBestAsk());
483         if (nbobetter && willTransact)
484             bestMarket = nbbo.getBestAskMarket();
485     } else { // sell
486         boolean nbobetter = nbbo.getBestBid() != null
487             && nbbo.getBestBid().greaterThanQuote.getBidPrice();
488         boolean willTransact = price.lessThan(nbbo.getBestBid());
489         if (nbobetter && willTransact)
490             bestMarket = nbbo.getBestBidMarket();
491     }
492
493     if (!bestMarket.equals(this))
494         log.info("Routing %s to %s from %s to NBBO %s",
495             agent, type, quantity, price, this, quote, bestMarket, nbbo);
496     bestMarket.submitOrder(agent, type, price, quantity, currentTime, duration);
497 }

```

Figure B.2: ZI order routing logic from MarketSim. Note that the code first checks that quantity is greater than zero, then uses whether quantity is greater than zero to determine whether the order is to buy or sell the asset. All orders will enter the ‘buy’ block, thus being routed based on the best ask price in the market even if the incoming order is a sell order.

In this section, we examine the results from including the apparent order routing bug in addition to the *MarketSim* greedy strategy logic. We will refer to this implementation as *BestGuess+MS+bug*, whereas our implementation that includes only the greedy strategy logic from *MarketSim* is referred to as *BestGuess+MS*. First, the quantitative alignment results for the ZI surplus from *BestGuess+MS+bug* versus the canonical results are shown in Table B.2. The quantitative alignment results for the LA surplus values are shown in Table B.3. If we compare these to the results from the main paper, we see that this model achieves closer quantitative alignment than *BestGuess* and *MarketSim*. The *BestGuess+MS+bug* results are similar to *BestGuess+MS*, but the *BestGuess+MS* implementation aligns more closely for most experiments. The exception is for *2M LA* with $\delta = 75$ in Environment 3, where we

do not reject quantitative alignment for the *BestGuess+MS+bug* LA surplus results. However, we view this alignment as potentially spurious given the *2M LA* results are too high for $\delta = 50$ and too low for $\delta = 100$. These results overall motivate our choice to focus on the *BestGuess+MS* implementation in the main paper, under the belief that the order routing bug should not be included in the model unless it can help explain the original results in a way other interpretations can not.

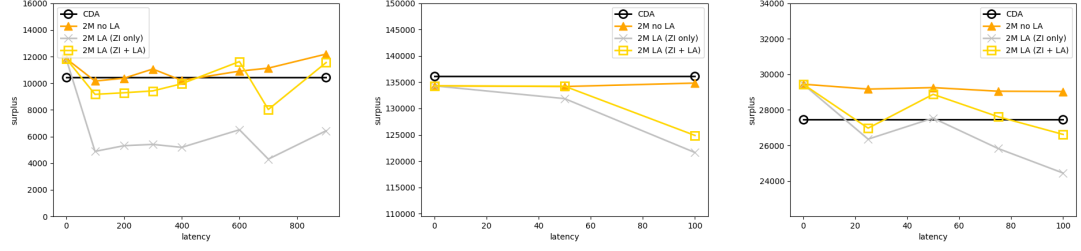
env	model	latency	WW	<i>BestGuess+MS+bug</i> Bootstrap ZI Surplus			
			mean	mean	SE	95% CI diff.	99% CI diff.
1	CDA	0	10383	10426.27	25.31	(-5.53, 93.25)	(-16.73, 109.80)
1	2M no LA	0	11807	11932.08	19.51	(85.79, 163.41)	(75.84, 171.66)
1	2M no LA	100	10373	9730.27	29.79	(-703.33, -587.89)	(-720.61, -574.97)
1	2M no LA	200	10621	10012.32	29.14	(-665.62, -553.62)	(-684.49, -538.06)
1	2M no LA	300	11244	10608.77	24.92	(-687.00, -588.50)	(-709.19, -572.06)
1	2M no LA	400	10438	9970.25	32.10	(-534.39, -409.98)	(-549.40, -389.80)
1	2M no LA	600	11128	10387.98	24.86	(-786.02, -689.18)	(-801.83, -673.52)
1	2M no LA	700	11302	10637.86	24.04	(-709.43, -616.75)	(-721.01, -596.78)
1	2M no LA	900	12358	11938.44	19.81	(-457.74, -380.17)	(-466.94, -369.29)
1	2M LA	100	5919	7204.15	34.56	(1220.68, 1355.07)	(1201.28, 1372.52)
1	2M LA	200	6358	7296.69	33.71	(871.47, 1002.68)	(853.95, 1030.35)
1	2M LA	300	6398	7395.82	32.47	(935.52, 1063.26)	(915.52, 1075.02)
1	2M LA	400	6130	7698.96	29.11	(1515.53, 1629.38)	(1501.61, 1644.35)
1	2M LA	600	7459	8639.58	26.16	(1129.21, 1231.83)	(1111.91, 1245.24)
1	2M LA	700	5256	6278.89	44.05	(933.03, 1104.25)	(903.35, 1127.32)
1	2M LA	900	6819	8631.55	27.30	(1760.36, 1863.58)	(1740.56, 1878.50)
2	CDA	0	136140	136131.67	63.79	(-135.82, 114.54)	(-181.92, 156.36)
2	2M no LA	0	134339	134381.44	65.82	(-90.28, 168.64)	(-136.82, 207.63)
2	2M no LA	50	135789	134638.56	65.76	(-1274.73, -1021.46)	(-1324.56, -979.64)
2	2M no LA	100	136542	135172.16	68.72	(-1498.56, -1239.63)	(-1534.73, -1188.09)
2	2M LA	50	133177	130579.29	82.55	(-2748.18, -2442.16)	(-2815.83, -2379.43)
2	2M LA	100	124012	122376.68	92.74	(-1820.76, -1455.35)	(-1863.14, -1410.87)
3	CDA	0	27482	27471.48	39.97	(-92.47, 64.16)	(-119.32, 91.08)
3	2M no LA	0	29424	29250.72	30.49	(-233.90, -117.28)	(-250.45, -101.19)
3	2M no LA	25	29347	29021.88	30.98	(-386.76, -261.71)	(-406.60, -243.15)
3	2M no LA	50	29479	29148.87	28.31	(-386.89, -277.74)	(-400.12, -259.08)
3	2M no LA	75	29271	28944.63	29.64	(-382.92, -268.07)	(-402.71, -252.81)
3	2M no LA	100	29277	28932.74	31.96	(-407.85, -284.66)	(-426.89, -262.98)
3	2M LA	25	26612	25752.47	40.81	(-938.51, -784.25)	(-963.98, -766.58)
3	2M LA	50	27953	27321.18	36.07	(-700.07, -562.18)	(-719.21, -540.43)
3	2M LA	75	26388	26180.24	40.57	(-287.09, -132.65)	(-317.42, -110.81)
3	2M LA	100	25070	25311.84	43.62	(155.05, 330.57)	(134.14, 351.05)

Table B.2: Test of quantitative alignment for our *BestGuess+MS+bug* implementation results versus WW. Each WW ZI value is the mean ZI surplus reported by WW [189, pp. 83, 85] for 500 mixtures (100 simulation runs each) sampled according to the welfare-maximizing strategy profile. Our results are calculated from 5,000 mixtures (100 runs each) based on the same strategy profile. From these, we construct 1,000 bootstrap samples of 500 mixtures each to produce confidence intervals for the mean difference. We report the mean and standard error of the 1,000 bootstrap sample means, along with the 95% and 99% confidence intervals for the difference between our bootstrap sample means and the reported WW mean. Confidence intervals that contain zero are shown in **bold**, indicating where quantitative alignment is not rejected.

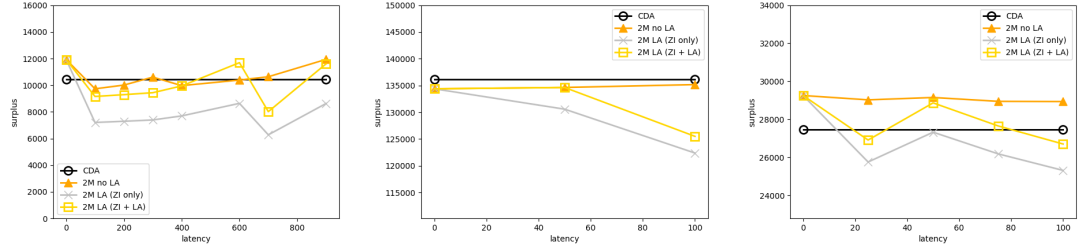
env	model	latency	WW LA	<i>BestGuess+MS+bug</i> Bootstrap LA Surplus			
			mean	mean	SE	95% CI diff.	99% CI diff.
1	2M LA	100	3487	1950.28	17.75	(-1571.17, -1501.88)	(-1581.64, -1492.61)
1	2M LA	200	3164	1997.30	18.13	(-1202.55, -1131.24)	(-1210.30, -1118.13)
1	2M LA	300	3224	2043.32	16.93	(-1213.13, -1148.59)	(-1221.61, -1141.54)
1	2M LA	400	4018	2250.88	18.27	(-1803.91, -1731.71)	(-1813.73, -1721.15)
1	2M LA	600	4349	3050.70	16.24	(-1330.56, -1266.93)	(-1339.43, -1260.81)
1	2M LA	700	2958	1737.36	16.56	(-1252.87, -1189.47)	(-1265.82, -1179.00)
1	2M LA	900	4825	2975.98	16.99	(-1881.00, -1814.80)	(-1893.96, -1805.96)
2	2M LA	50	2417	4049.84	43.97	(1550.15, 1716.65)	(1521.51, 1750.46)
2	2M LA	100	2888	3121.23	37.28	(163.52, 312.72)	(137.37, 330.55)
3	2M LA	25	538	1156.56	13.64	(592.06, 645.30)	(585.33, 654.13)
3	2M LA	50	1154	1542.84	16.44	(356.84, 422.29)	(347.76, 431.54)
3	2M LA	75	1470	1469.18	17.12	(-32.46, 34.18)	(-44.31, 43.86)
3	2M LA	100	1763	1387.12	16.28	(-406.43, -343.77)	(-417.74, -332.81)

Table B.3: Test of quantitative alignment of LA surplus results from our *BestGuess+MS+bug* replication versus WW. Each WW LA value is the mean LA surplus reported by WW [189, pp. 83, 85] for 500 mixtures (100 simulation runs each) sampled according to the welfare-maximizing strategy profile. Our results are calculated from 5,000 mixtures (100 runs each) based on the same strategy profile. From these, we construct 1,000 bootstrap samples of 500 mixtures each to produce confidence intervals for the mean difference. We report the mean and standard error of the bootstrap sample means, along with the 95% and 99% confidence intervals for the difference between our bootstrap sample means and the reported WW mean. Confidence intervals that contain zero are shown in **bold**, indicating where quantitative alignment is not rejected.

We present the graphical results from the two implementations along with the WW results in Figs. B.3, B.4, B.5, and B.7. We overall see similar behavior between these implementations. Most results are relationally equivalent to each other, while the cases with differences (e.g. the mean execution times in Environment 3, Fig. B.4) favor *BestGuess+MS* for more closely aligning to the original results.



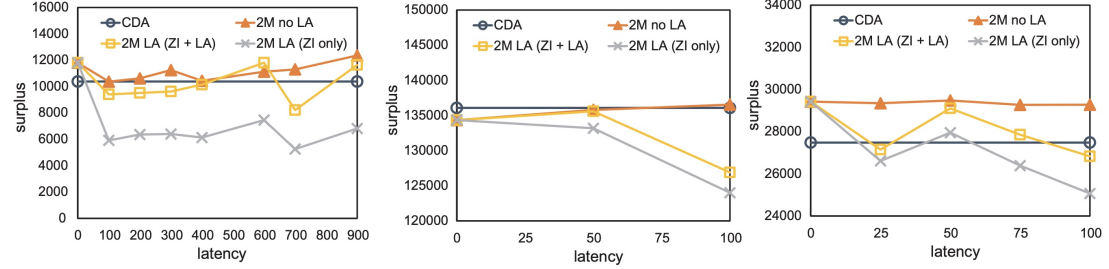
(a) Env. 1—BestGuess+MS (b) Env. 2—BestGuess+MS (c) Env. 3—BestGuess+MS



(d) Env. 1—
BestGuess+MS+bug

(e) Env. 2—
BestGuess+MS+bug

(f) Env. 3—
BestGuess+MS+bug

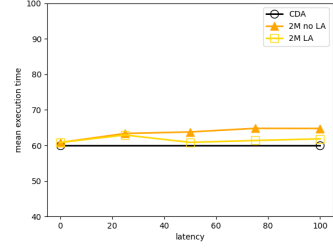
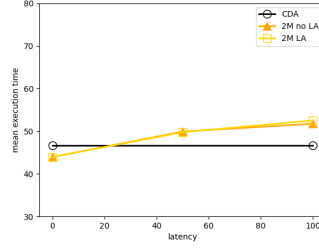
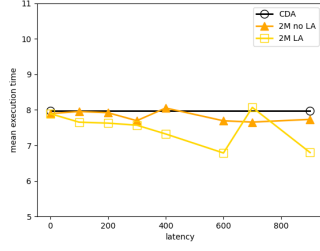


(g) Env. 1—WW

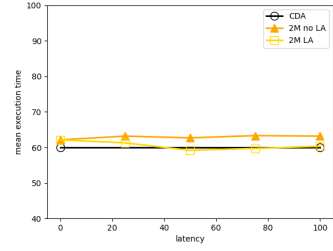
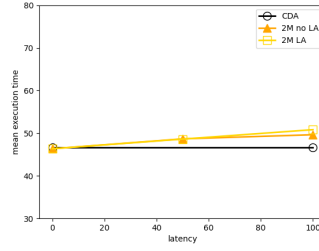
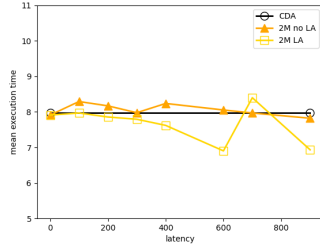
(h) Env. 2—WW

(i) Env. 3—WW

Figure B.3: Mean surplus results from our BestGuess+MS and BestGuess+MS+bug implementations of the model versus figures reproduced from WW [189, p. 82]. The WW figures show the mean surplus from 500 mixtures, 100 runs per mixture. Our results show the mean surplus from 5,000 mixtures, 100 runs per mixture.



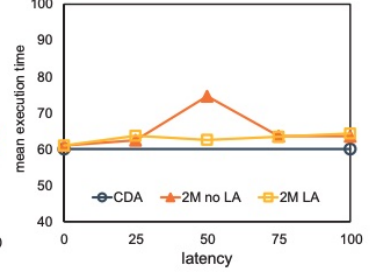
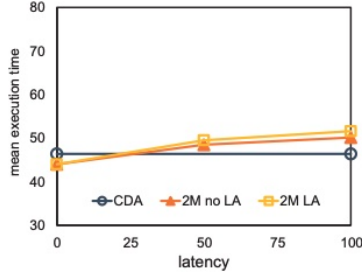
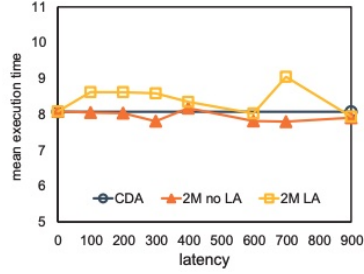
(a) *Env. 1—BestGuess+MS* (b) *Env. 2—BestGuess+MS* (c) *Env. 3—BestGuess+MS*



(d) *Env. 1—
BestGuess+MS+bug*

(e) *Env. 2—
BestGuess+MS+bug*

(f) *Env. 3—
BestGuess+MS+bug*

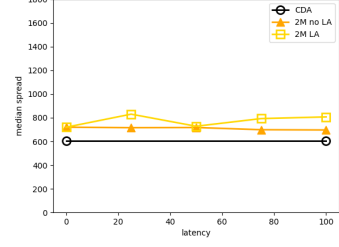
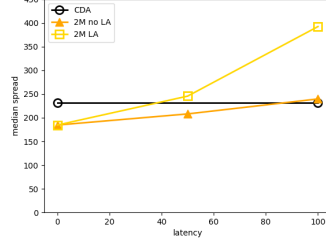
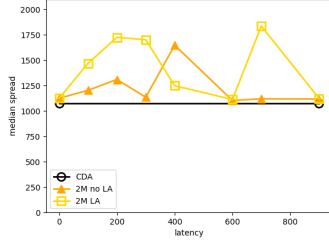


(g) *Env. 1—WW*

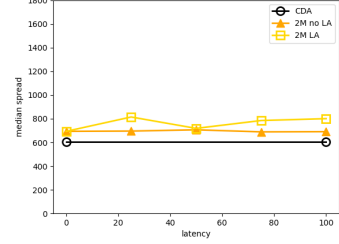
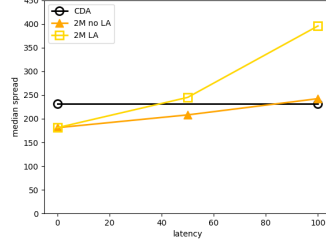
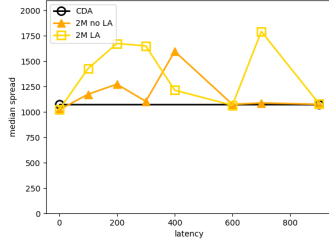
(h) *Env. 2—WW*

(i) *Env. 3—WW*

Figure B.4: Mean execution time results from our *BestGuess+MS* and *BestGuess+MS+bug* implementations of the model versus figures reproduced from WW [189, p. 86]. The WW figures show the mean execution time from 500 mixtures, 100 runs per mixture, for each experiment. Our results show the mean execution time from 5,000 mixtures, 100 runs per mixture.



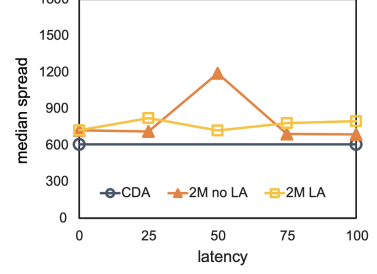
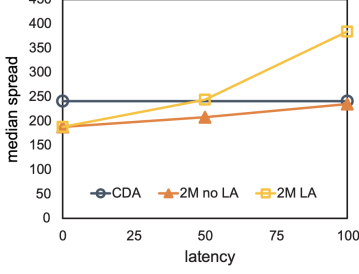
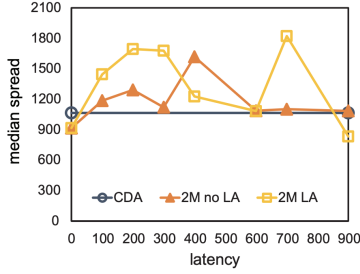
(a) Env. 1—BestGuess+MS (b) Env. 2—BestGuess+MS (c) Env. 3—BestGuess+MS



(d) Env. 1—
BestGuess+MS+bug

(e) Env. 2—
BestGuess+MS+bug

(f) Env. 3—
BestGuess+MS+bug

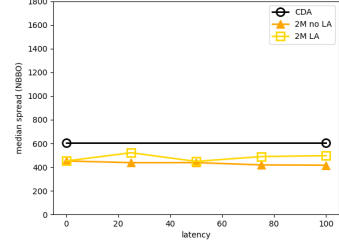
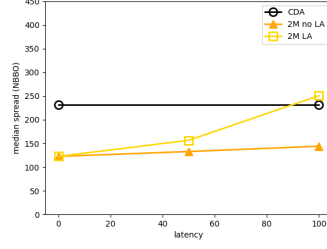
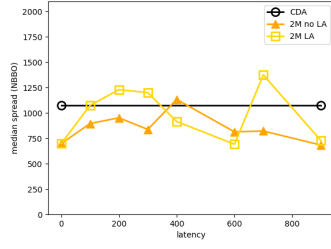


(g) Env. 1—WW

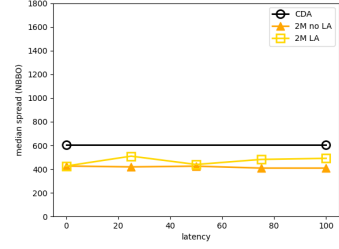
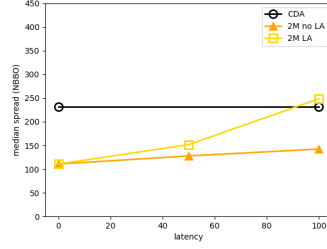
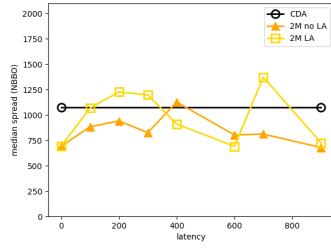
(h) Env. 2—WW

(i) Env. 3—WW

Figure B.5: Mean median spreads from the exchange BBOs for our BestGuess+MS and BestGuess+MS+bug implementations of the model versus the figures reproduced from WW [189, p. 87]. BBO spreads are averaged across the different exchanges in a given simulation run. Results from our simulation are averaged over 5,000 mixtures, 100 runs per mixture, for each experiment.



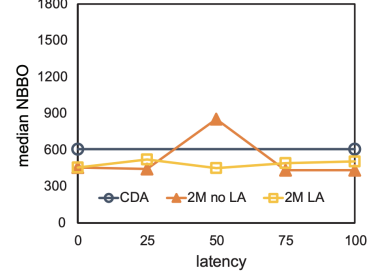
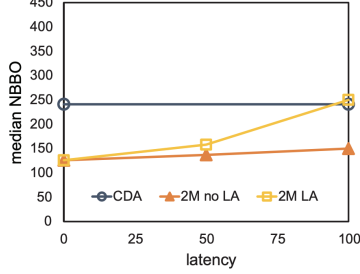
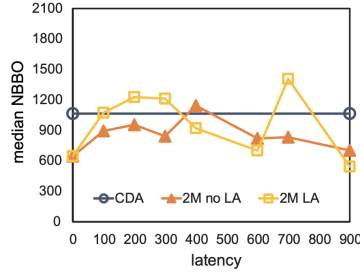
(a) Env. 1—BestGuess+MS (b) Env. 2—BestGuess+MS (c) Env. 3—BestGuess+MS



(d) Env. 1—
BestGuess+MS+bug

(e) Env. 2—
BestGuess+MS+bug

(f) Env. 3—
BestGuess+MS+bug



(g) Env. 1—WW

(h) Env. 2—WW

(i) Env. 3—WW

Figure B.6: Mean median SIP NBBO spread for our BestGuess+MS and BestGuess+MS+bug implementations of the model versus the figures reproduced from WW [189, p. 87]. Results from our simulation are averaged over 5,000 mixtures, 100 runs per mixture, for each experiment.

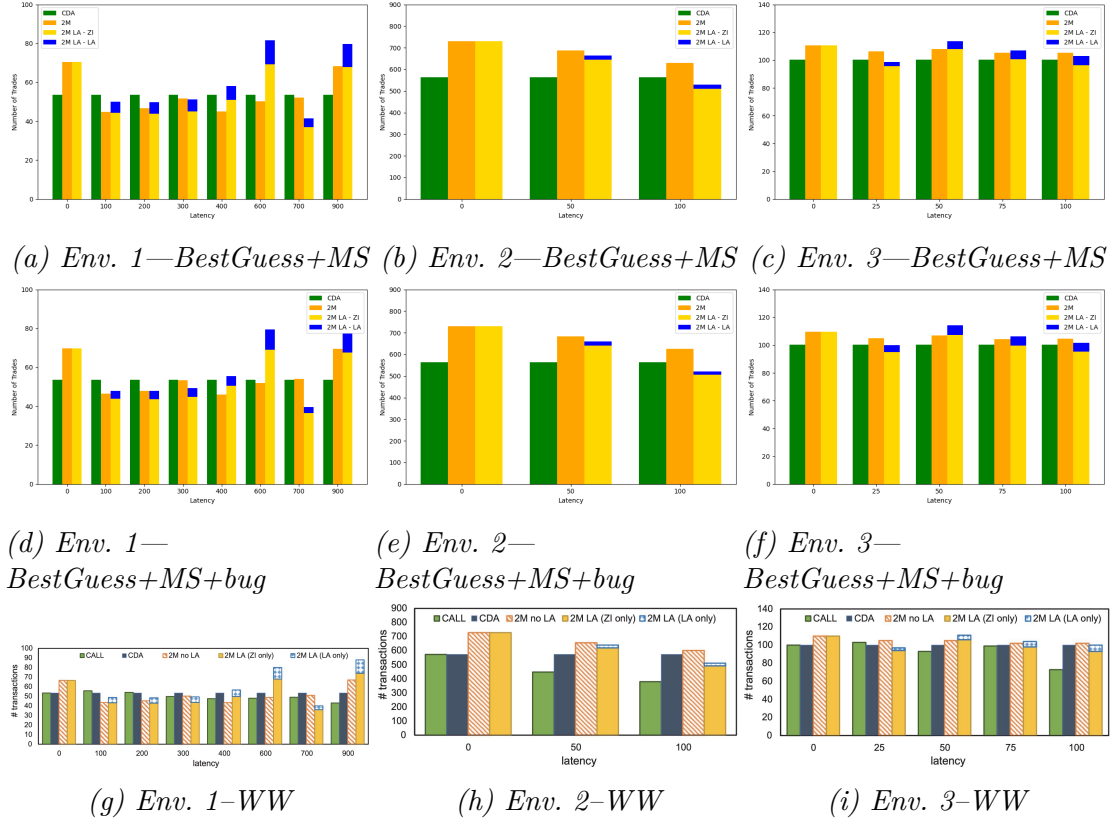


Figure B.7: Mean number of transactions from our BestGuess+MS and BestGuess+MS+bug implementations of the model versus the figures reproduced from WW [189, p. 86]. Please note, a single trade results in two transactions for this measurement (one per trader involved in the trade). Results from our simulation are averaged over 5,000 mixtures, 100 runs per mixture, for each experiment.

B.3 ODD PROTOCOL

Below, we give the model description, which follows the ODD (Overview, Design concepts, Details) protocol [90, 91] for our replication of Wah and Wellman’s 2016 model (WW) [189]. We follow as closely as possible the model as specified by the text of WW, and we include annotations where appropriate for specific pieces of logic. We also make note of any ambiguities from the text for which we needed to make

a specific design decision (e.g. specifically how a number drawn from a continuous distribution is converted into an integer).

B.3.1 OVERVIEW

Purpose

This model examines the effects of fragmentation, latency, and latency arbitrage in a financial market. The model features a single stock traded by investor agents in a market with one or more stock exchange. A consolidated market with a single exchange is modeled first, after which an additional exchange, communication latency, and a latency arbitrage agent are incrementally added to examine their respective effects. The hypothesis is that unequal access to cross-market information impacts the efficient allocation of value to investors and the liquidity of the market overall.

Entities, state variables, and scales

Entities	Scheduler: schedules and kicks off events in the simulation Exchange: market operating a continuous double auction (CDA) Security: fungible asset traded on each exchange Security Information Processor (SIP): data consolidator for the market; maintains and publishes the National Best Bid and Offer (NBBO) Traders: Zero Intelligence (ZI) background traders Latency Arbitrage (LA) trader
-----------------	--

Table B.4: Entities in the model ODD protocol.

State variables	Scheduler
	Current time t
	List of scheduled events (ZI arrivals and SIP updates)
	Exchange
	Limit order book (LOB): all resting orders with prices, quantities, and timestamps
	Best bid and offer (BBO): highest buy price and lowest sell price on the exchange
	BBO subscribers
	SIP
	Exchange BBO(s): Current BBO feeds from the exchange(s)
	National Best Bid and Offer (NBBO)
	NBBO subscribers
	Security
	Fundamental value r_t
	ZI trader
	Valuation vector Θ_i : marginal private benefits for buying and selling the asset
	Strategy parameters $\{R_{\min}, R_{\max}, \eta\}$
	Primary exchange
	Primary exchange BBO
	Current holdings: quantity q_i the trader is long or short
	Outstanding orders
	Profits from trading
	LA trader (if present)
	Exchange BBOs
	Profits from trading

Table B.5: State variables by entity in the model ODD protocol.

Scales	T : Simulation length $N_{exch} \in \{1, 2\}$: Number of exchanges $\delta \geq 0$: SIP latency N_{ZI} : Number of ZI traders $N_{LA} \in \{0, 1\}$: Presence or absence of the LA trader λ : ZI arrival rate $q_{\max}$: maximum shares ZI trader can be long or short the asset
---------------	---

Table B.6: Scales in the model ODD protocol.

Process overview and scheduling

Time is discrete, with steps $t \in 1, \dots, T$. A scheduler maintains a queue of exogenous events, ordered by time and then by insertion order (first-in, first-out). Only two types of events go through the scheduler queue: ZI trader arrivals and SIP NBBO update events (scheduled with latency δ after an exchange BBO change). All other actions in the model are triggered immediately (with zero modeled latency) as consequences of scheduled events. Non-scheduled actions include order submissions and cancellations, exchange matching and trade execution, exchange BBO updates, LA trader responses, and notifications of trades.

B.3.2 DESIGN CONCEPTS

Basic principles

Underlying the model is the concept that fragmentation in a market leads to communication latency. This latency affects different classes of traders unequally. Retail and institutional investors are assumed to trade based on some understanding of the fundamental value of the security and additional private factors. These investors are

less invested in speed technologies than the HFT traders and therefore have some exposure to stale information in the market relative to HFT traders. This is represented in the model through a simplified latency arbitrage scenario, with the SIP connecting the markets representing possibly stale information (when $\delta > 0$) relative to the direct access the HFT arbitrage agent has to prices at each exchange. These assumptions (fragmentation, speed/information differentials, and high-level trading strategies) determine the structure and individual behaviors in the model, and the model provides insight into the impact these features have on the system overall through various market-level performance characteristics.

Emergence

The key results examined by WW are the surplus achieved by traders, mean execution time of orders that result in trades, the number of transactions, and the median spread between bid and ask prices in the market. These features are emergent from the interactions of traders adding orders in the simulation.

Adaptation

No details/parameters of the ZI or LA strategies change over the course of the simulation in order to improve profitability.

Objectives

Traders aim to maximize their surplus (private value plus profit), but they are not strategic beyond what they are hard-coded to do. The ZI traders aim to achieve immediate execution if possible through their routing strategy, which would serve to

increase its surplus. The LA agent submits orders immediately at prices ensured to execute and achieve immediate profit

Learning

The agents do not learn or adapt their traits over the course of the simulation.

Prediction

No prediction is done by the agents.

Sensing

Traders use data feeds in the market to sense the prices available. This determines whether to act in the LA agent's case, and it determines the order routing decision for the ZI agents.

Interaction

Traders interact through the matching of their orders in the LOBs at the exchanges. The resting orders at each exchange inform the routing decisions of the subsequent traders placing orders. Trader actions on the LOBs are communicated via BBO and NBBO updates, which are received as they are updated. These updates occur instantly for the BBOs and with latency δ for the NBBO after events in the model.

Stochasticity

The fundamental price of the security is a mean-reverting stochastic process. ZI trader's internal valuations and order prices are assumed to be stochastic perturba-

tions on top of the security’s fundamental value. Exact arrival times of ZI agents in the market are also stochastic, with interarrival times drawn from an exponential distribution.

Collectives

Equal sized groups of ZI traders are associated with each of the exchanges. This association determines which exchange’s direct BBO feed the trader receives and determines the trader’s default routing decision. These features in turn factor into the trader’s specific actions. Aside from this difference between groups of ZI traders, there are no collectives in the model.

Observation

The evolution of the order books, including trades executed in the market, are logged by the exchange BBOs and the SIP NBBO. Every time an order is submitted to an exchange, this results in a trade or an order added to the LOB. As a result, the exchange’s BBO and the SIP NBBO are updated. Trades are tracked as a global variable that is updated by the exchange(s) any time a trade occurs. The trade, BBO, and NBBO records are therefore updated in event-time. These event-time observations are used at the end of the simulation to calculate the performance metrics, which are then logged.

B.3.3 INITIALIZATION

Before each model run, the traders, exchanges, and SIP are instantiated. First, we create the exchanges and the SIP. The SIP’s NBBO and exchange BBO(s) are

initialized to have null values as their best bid and best offer. The SIP subscribes to the direct BBO feed from each exchange.

The security traded in the market is also initialized. We pre-compute the security’s fundamental value series $\{r_t\}$ for each integer time point in the simulation (i.e. $t \in [1, T]$) using the logic detailed for the security (Eq. 3.1). The calculation of $\{r_t\}$ at initialization is for computational efficiency; it could be calculated at each time step, and we do not believe this would impact the model.

ZI initialization

Next, we create the ZI traders. Each ZI trader is assigned one of the markets as its primary market. In the single-exchange configuration of the market, all ZI traders are assigned to the only exchange. In the two-market configurations, we alternate which exchange an agent is assigned to each time we create an agent. We therefore evenly split the ZI traders between the two exchanges, as there are no environments that feature an odd number of ZI agents. Each ZI agent is assigned trading strategy parameters according to the specified mixture probabilities, specified in Table B.10. We sample ZI trader profiles according to the surplus-maximizing equilibrium mixture probabilities provided by WW for the given experiment (environment + latency + number of LA). The surplus-maximizing equilibrium mixture probabilities are determined by combining the information in Tables 6–11 in WW [189, pp. 83–86]. Let a ‘mixture’ refer to a sample of N_{ZI} profiles according to the mixture probabilities. A mixture is stochastically drawn based on the probabilities for a given run. Each profile from the mixture is assigned to one of the ZI agents. ZI agents subscribe to the BBO feed from their assigned primary exchange and to the NBBO feed from the

SIP. ZI agents initialize their marginal private benefits vector Θ according to the logic detailed in Eq. B.2.

Schedule initialization

The Scheduler is initialized, with initial arrivals for the ZI agents scheduled. WW say, “The background traders arrive at the market according to a Poisson process with rate λ ” [189, p. 75]. They do not specify how the ZI arrival times are made to be discrete. We draw the ZI arrival times independently from an exponential distribution ($[Exponential(1/\lambda)]$). We choose to take the ceiling such that no event will be scheduled at time $t = 0$. We account for this by letting arrivals occur on times 1 through T (inclusive), giving T total time steps.

LA initialization

Finally, if the given model configuration includes an LA agent, it is also initialized. Mainly, the LA agent subscribes to both exchanges’ BBO feeds.

B.3.4 INPUT DATA

The model does not use input data aside from the ZI strategy mixture probabilities (discussed above) and the parameters, which are specified in the next subsection.

B.3.5 PARAMETERS

The three environments specified by WW are given in Table B.7. The parameter N_{ZI} is the number of ZI traders, λ is the arrival rate of ZI traders, κ is the strength of

mean reversion of the security’s fundamental value, T is the simulation length, and Δ_δ is the granularity of latency steps across experiments for a given environment. The parameters that are fixed across environments are shown in Table B.8.

Environment	N_{ZI}	λ	κ	T	Δ_δ
1	24	0.05	0.05	15,000	100
2	238	0.005	0.02	10,000	10
3	58	0.005	0.02	5,000	10

Table B.7: Parameter settings for the three market environments from WW [189, p. 80]. Please note, the λ parameter for Environments 1 and 2 has an apparent typo ($\lambda = 0.0005$) in WW’s Table 3 [189, p. 80]; the 0.005 value is specified in the text [189, p. 80] and in Wah’s dissertation [188].

Parameter	Default	Description
r_{mean}	100,000	Mean fundamental value of security.
σ_{shock}^2	5,000,000	Variance of the random shock to the security’s fundamental value.
σ_{PV}^2	5,000,000	Variance of ZI private valuations; used to draw private marginal utility of the agent’s net position.
α	0.001	Threshold from which the LA agent determines whether an opportunity is worth pursuing.
q_{max}	10	Maximum position (long or short) for ZI traders

Table B.8: Default parameters used across model configurations. These were specified by WW [189, p. 80] except for α . The α value is based on the 2013 paper [190] and Wah’s dissertation [188], and it is further confirmed by looking at the default value α in MarketSim.

ID	$R_{\min}$	$R_{\max}$	η
ZI_1	0	125	1
ZI_2	0	250	1
ZI_3	0	500	1
ZI_4	250	500	1
ZI_5	0	1000	1
ZI_6	500	1000	0.4
ZI_7	500	1000	1
ZI_8	0	1500	0.6
ZI_9	1000	2000	0.4
ZI_{10}	0	2500	0.4
ZI_{11}	0	2500	1

Table B.9: ZI strategy combinations, taken from WW [189, p. 80].

Env	Model	Latency	ZI_1	ZI_2	ZI_3	ZI_4	ZI_5	ZI_6	ZI_7	ZI_8	ZI_9	ZI_{10}	ZI_{11}
1	CDA	-	0	0	0	0	0	0	0	0	0.507	0.493	0
1	2M no LA	0	0	0	0	0	0	0	0	0	0	1.0	0
1	2M no LA	100	0	0	0	0	0	0.602	0	0	0.239	0.159	0
1	2M (LA)	100	0	0	0	0	0	0.237	0	0	0.537	0.226	0
1	2M no LA	200	0	0	0	0	0	0.381	0	0	0.338	0.281	0
1	2M (LA)	200	0	0	0	0	0	0	0	0	0.679	0.321	0
1	2M no LA	300	0	0	0	0	0	0.692	0	0	0.036	0.272	0
1	2M (LA)	300	0	0	0	0	0	0	0	0	0.655	0.345	0
1	2M no LA	400	0	0	0	0	0	0	0	0	0.595	0.405	0
1	2M (LA)	400	0	0	0	0	0	0.47	0	0	0.258	0.272	0
1	2M no LA	600	0	0	0	0	0	0.81	0	0	0	0.19	0
1	2M (LA)	600	0	0	0	0	0	0	0.029	0	0	0.971	0
1	2M no LA	700	0	0	0	0	0	0.739	0	0	0	0.261	0
1	2M (LA)	700	0	0	0	0	0	0.006	0	0	0.826	0.168	0
1	2M no LA	900	0	0	0	0	0	0	0	0	0	1.0	0
1	2M (LA)	900	0	0	0	0	0	0.131	0	0	0	0.869	0
2	CDA	-	0	0	0	0	0	0	0	0	0	0.659	0.341
2	2M no LA	0	0.146	0	0	0	0	0	0	0	0	0.854	0
2	2M no LA	50	0	0.162	0	0	0	0	0	0	0	0.832	0
2	2M (LA)	50	0	0	0.188	0	0	0	0	0	0	0.812	0
2	2M no LA	100	0.051	0	0	0	0	0	0	0	0	0.76	0.189
2	2M (LA)	100	0	0	0	0	0	0	0	0	0.233	0.767	0
3	CDA	-	0	0	0	0	0	0	0	0	0.248	0.752	0
3	2M no LA	0	0	0	0.017	0	0	0	0	0	0.004	0.979	0
3	2M no LA	25	0	0	0	0	0	0	0	0	0	0.854	0.146
3	2M (LA)	25	0	0	0	0	0	0	0	0	0.21	0.79	0
3	2M no LA	50	0	0	0	0	0	0	0	0	0	0.948	0.052
3	2M (LA)	50	0	0	0	0	0	0	0	0.065	0.043	0.892	0
3	2M no LA	75	0	0	0	0	0	0	0	0	0	0.823	0.177
3	2M (LA)	75	0	0	0	0	0	0	0	0	0.142	0.858	0
3	2M no LA	100	0	0	0	0	0	0	0	0	0	0.839	0.161
3	2M (LA)	100	0	0	0.015	0	0	0	0	0	0.231	0.754	0

Table B.10: ZI surplus-maximizing equilibrium mixtures for each experiment (environment+configuration+latency). ZI strategy settings (ZI_i) are as specified in Table B.9.

B.3.6 SIMULATION DRIVER

The simulation driver initializes the simulation by creating the scheduler, exchanges, traders, security, and SIP. It kicks off the initialization functions for each of these according to the logic detailed in the Initialization section. This results in a population of exchanges, traders, a security with a time-dependent fundamental value, and data feeds in the market. The simulation driver then kicks off the schedule to run through events. The scheduler adds events along the way as messages are sent by agents and as ZI agents reschedule themselves for arrival. When the scheduler finishes all events, the simulation driver kicks off the functions to save the data from the simulation for analysis.

B.3.7 SCHEDULER

The scheduler maintains the evolving schedule of events, kicking them off based on their time. WW say, “[E]vents are maintained in a queue ordered by time of occurrence” [189, p. 78], and Wah’s dissertation says, “Multiple events may be scheduled for the same time step, in which case they are executed deterministically in the order in which they are enqueued” [188, pp. 16–17]. We therefore assume events are prioritized first by time, then by order of scheduling (i.e., first-in-first-out for each time step).

An event that has sub-events will kick off those related actions directly instead of going through the scheduler. When the simulation is started, the scheduler kicks off each event based on its time and the order in which it was enqueued for the current time step. Once there are no more events at the current time step, the scheduler skips

ahead to the next time step which has a scheduled event. No events are scheduled for $t > T$.

B.3.8 EXCHANGES

Each exchange in the model operates a continuous double auction (CDA)² for the asset. A CDA is a two-sided market where agents submit orders to buy and sell the asset. The market is ‘continuous’ in the sense that orders and trades occur as soon as they are received and processed by the exchange’s matching engine. A limit order to buy (sell) the stock specifies a maximum (minimum) amount the agent is willing to trade at and a quantity to buy (sell). In the model, all orders are for a single unit of the stock. Each exchange maintains a limit order book (LOB), constructed with the following elements:

1. Orders: map from order IDs to their respective orders
2. Bids: map from price to the queue of buy order IDs at that price
3. Asks: map from price to the queue of sell order IDs at that price

The exchange publishes a BBO to its subscribers, a feed containing the best bid and offer prices in the LOB at that moment. LOBs are initialized with no bids or asks at the start. The BBO has null prices for its best available bid and ask whenever their respective side of the LOB is empty (including at initialization).

When an exchange receives an add message for order O_i with limit price p , it does the following. First, it checks whether the added price can be matched against

²Wah and Wellman [190, 189] also examined an alternative clearing mechanism, periodic call auctions, as a possible remedy to latency arbitrage. All exchanges in the U.S. stock market currently use CDAs, however, and we leave the periodic call auction models out of scope for now.

existing orders in the book. More explicitly, if order O_i is a buy order, the exchange checks whether there is a resting sell order O_j with price $q \leq p$. Similarly, if O_i is a sell order, the exchange checks whether there is a resting buy order O_j with price $q \geq p$. In either of these cases, the orders are matched at price q , resulting in a transaction. If multiple resting orders exist on the LOB that could match with the incoming order, the oldest order is given priority (i.e. price-time priority). In this simulation, all orders are for a single quantity of the asset, and so each trade is always for a single unit. Receipts are sent to the traders associated with i and j , and O_j is removed from the LOB. If O_i instead does *not* immediately match, it is added to the LOB. A trader can request to withdraw a resting order from the LOB. Upon receiving a withdraw request, the exchange removes the corresponding order from the LOB.

After any change to the exchange’s LOB due to a withdraw, add, or trade, the exchange publishes its updated best bid and ask prices to its subscribers. This is published regardless of whether the best bid or ask prices were changed by the LOB update. The exchange loops over its subscribers in the order they subscribed to the exchange’s BBO feed, directly calling the subscriber’s `update_bbo()` function.

The market can be configured with one or two exchanges. Each ZI trader agent has a ‘primary’ exchange. In the two-market configurations, the ZI agents are split evenly between the two exchanges and subscribe only to the BBO feed of their primary exchange. The SIP subscribes to both exchange feeds, as does the latency arbitrage (LA) agent if it is present.

B.3.9 THE SIP

The role of the SIP in the simulation is to consolidate information from across the fragmented market and publish the national best bid and offer (NBBO). This process is kicked off whenever the SIP receives a BBO update from an exchange. We assume that when latency δ is zero the SIP should update its NBBO (and subscribers) without going through the scheduler. In our implementation, this means when the SIP receives an `update_bbo()` call from an exchange and $\delta = 0$, the SIP directly calls its own `update_nbbo()` function. This prevents agents acting on the same time step from having stale information due to the SIP being queued behind their already-scheduled action. When $\delta > 0$, we assume an `update_nbbo()` call is scheduled at time step $t + \delta$ using the Scheduler, with no priority preference. In other words, if a ZI agent is already scheduled for that time step, the NBBO update will occur after that agent makes its trading action (and any resultant actions).

When `update_nbbo()` is called, the NBBO is calculated by finding the best BID (ASK) at one exchange and then seeing whether the next exchange has a higher (lower) BID (ASK). If neither exchange has a BID (ASK) order, the NBBO BID (ASK) is empty. WW do not specify how ties for the best BID or ASK should be handled. We arbitrarily award ties to the first exchange in the list. NBBO updates specify the time t of the update, the best BID price and exchange, and the best ASK price and exchange. Once calculated, NBBO updates are published to the subscribers of the SIP in the order in which they subscribed. This is done by looping through the subscribers and directly calling each subscriber's `update_nbbo` function.

B.3.10 SECURITY

The security being traded in the market is mostly abstract, except it possesses an evolving ‘common fundamental value’ known by the traders. As noted in the Initialization section, we pre-compute the fundamental value series $\{r_t\}$ for efficiency reasons. This series is calculated following the below logic:

```

1  values = {}
2  r = r_mean
3  for i in [1,...,T]:
4      shock = N(0, sigma_shock^2)
5      r = max(0, kappa * r_bar + (1-kappa) * r + shock)
6      fundamental_value[i] = r

```

The mean is r_{mean} , the degree of mean-reversion is κ (**kappa**), and $\mu_t \sim N(0, \sigma_{\text{shock}}^2)$ is a Gaussian random shock drawn independently for each time step.

WW do not specify what r_0 should be, so we assume $r_0 = r_{\text{mean}}$. WW also do not specify how or whether the fundamental value should be made to be an integer (since prices are discrete in the model). We choose not to discretize the fundamental value when it is being calculated. A ZI trader arriving at time t looks up r_t as part of its trading strategy. We assume r_t should be rounded to the nearest integer whenever it is looked up by a trader.

The ZI traders use the estimated terminal value of the security in their trading strategy. We model this logic as a function of the security, which the ZI agents call. The estimate is based on the fundamental value r_t at the time t the trader looks it up. The estimated terminal value r_t^{est} is then the expected fundamental value at time T , accounting for the mean-reversion process. WW do not specify how r_t^{est} is made to be an integer, nor whether each component to the price should be converted to int or just the final price. We assume that only the estimated terminal value should be

rounded after the calculation of r_t^{est} . In other words,

$$r_t^{\text{est}} = \text{round} \left(\left(1 - (1 - \kappa)^{T-t} \right) r_{\text{mean}} + (1 - \kappa)^{T-t} r_t \right). \quad (\text{B.1})$$

B.3.11 ZERO INTELLIGENCE (ZI)

ZI Initialization

As noted in the Initialization section, each ZI trader subscribes to the SIP NBBO and to the BBO feed from its primary exchange. Its holdings, profits, and number of transactions are all initialized to zero. Next, for the given ZI trader ZI_i , we construct Θ_i , the vector of differences in private benefits of changes to the trader's net position:

$$\Theta_i = \{\theta_i^q\}, \quad (\text{B.2})$$

where $-q_{\text{max}} < q \leq q_{\text{max}}$ and the θ_i 's are sorted such that $\theta^q \geq \theta^{q+1} \forall q$. No discretization logic is specified by WW, nor do they state anywhere that the internal component for the ZI agents should be integers. Hence, we do not round or otherwise discretize the θ values when they are initialized.

Arrival and base trading strategy

The ZI traders are scheduled to initially arrive at the market on a fixed schedule drawn from a Poisson distribution, as detailed in the Initialization section. Each ZI trader arrives to place an order at least once. We assume the first thing the ZI trader ZI_i does upon arrival at time t is schedule its next arrival for time $t + \lceil b_i \rceil$, where $b_i \sim \text{Exponential} \left(\frac{1}{\lambda} \right)$. WW do not specify how ZI arrival times should be made

discrete. We choose to take the ceiling so the trader does not schedule itself to arrive again on the same time step.

After scheduling its next arrival, ZI_i withdraws any outstanding orders from previous turns that have not yet resulted in a trade. It then determines its trading action for this turn. It randomly chooses (with equal probability) whether to buy or sell on this turn. Traders are not allowed to have a net position of $q_{\max}$ shares long or short of the asset. Thus, if the decision to trade or buy would result in a net position $|q| > q_{\max}$, the trader does not submit an order and exits.

Otherwise, ZI_i looks up the estimated terminal fundamental value r_t^{est} of the security (see the Security submodel section). It then calculates its internal valuation $v_i(t)$ of the asset based on r_t^{est} and its marginal private benefits vector Θ_i :

$$v_i(t) = r_t^{\text{est}} + \begin{cases} \theta_i^{q_i+1} & \text{if buying one unit} \\ \theta_i^{q_i} & \text{if selling one unit} \end{cases}. \quad (\text{B.3})$$

We choose to round $v_i(t)$ to the nearest integer, as WW do not specify how prices are made to be discrete. ZI_i 's order price p is 'shaded' some amount away from its private valuation in order to achieve surplus value.

$$p \sim \begin{cases} \max(0, U[v_i(t) - R_{\max}, v_i(t) - R_{\min}]) & \text{if buying} \\ \max(0, U[v_i(t) + R_{\min}, v_i(t) + R_{\max}]) & \text{if selling} \end{cases}, \quad (\text{B.4})$$

where U here denotes an integer uniformly chosen at random within the specified range. The requested surplus for ZI_i is then $s_i(t) = |v_i(t) - p_i(t)|$.

Before submitting its order, ZI_i determines whether there is a greedy opportu-

nity based on p , the trader's η parameter, and the current market information. As detailed in the main paper, we assess two interpretations of this greedy strategy; one implementation based on our 'best guess' based on the text, and one implementation based on *MarketSim*. We detail these two implementations below.

***BestGuess* greedy strategy**

WW say the ZI agents have access to quotes on their primary market and the NBBO. Let **NBBO** denote the current $NBBO_t$ received from the SIP as of time t , and let **BBO** denote ZI_i 's primary exchange BBO_t as of time t . Based on these sources, ZI_i determines the best bid price and the best ask:

```

1     max_bid = NBBO.bid.price
2     min_ask = NBBO.ask.price.
3
4     if BBO.bid != None and (max_bid == None or BBO.bid >= max_bid):
5         max_bid = bbo.bid
6     if BBO.ask != None and (min_ask == None or BBO.ask <= min_ask):
7         min_ask = BBO.ask

```

ZI_i determines whether there is a greedy opportunity based on p , the trader's η parameter, and **max_bid** and **bid_ex** if selling (or **min_ask** and **ask_ex** if buying). If so, ZI_i will update its order price p to the observed available price it is trying to immediately execute at.

```

1     requested_surplus = |valuation - p|
2     if side == 'buy':
3         if min_ask != None and
4             (requested_surplus * eta <= valuation - min_ask):
5             p = min_ask
6     else: // 'sell'
7         if max_bid != None and
8             (requested_surplus * eta <= max_bid - valuation):
9             p = max_bid

```

MarketSim greedy strategy:

In the *MarketSim* implementation of the greedy strategy, ZI_i does not look at the NBBO when determining whether there is a greedy opportunity, and it updates $p = v_i(t)$ when there is a greedy opportunity:

```
1     requested_surplus = |self.valuation - price|
2     if side == 'buy':
3         if BBO.ask != None and
4             requested_surplus * eta <= valuation - BBO.ask:
5             price = valuation
6     else: // 'sell'
7         if BBO.bid != None and
8             requested_surplus * eta <= BBO.bid - valuation:
9             price = valuation
```

Order submission

After the price is finalized (for either greedy strategy implementation), the order is then submitted to the exchange determined by the below logic. Assume `order.price` is set to p and `order.exchange` is set to ZI_i 's primary exchange. Also assume `BBO` is ZI_i 's primary exchange's current BBO and `NBBO` is the current NBBO received from the SIP.

```
1     route_order(order):
2         NBBO_price_better = False
3         will_transact = False
4         if order.type == "BUY":
5             if NBBO.ask.price != Null:
6                 if (BBO.ask == Null) or (NBBO.ask.price < BBO.ask):
7                     NBBO_price_better = True
8                 if order.price >= NBBO.ask.price:
9                     will_transact = True
10            alt_ex = NBBO.ask.exchange
11        elif order.type == "SELL":
12            if NBBO.bid.price != Null:
13                if (BBO.bid == Null) or (NBBO.bid.price > BBO.bid):
14                    NBBO_price_better = True
15                if order.price <= NBBO.bid.price:
```

```

16         will_transact = True
17         alt_ex = NBBO.bid.exchange
18
19         if NBBO.price_better and will_transact:
20             order.exchange = alt_ex
21
22         order.exchange.submit_order(order)

```

Calculation of surplus

At the end of the simulation, we calculate the total surplus S_i for an agent ZI_i with a terminal net position q_i^T and valuation vector $\Theta_i = \{\theta_j\}$. This is calculated as the sum of the following: the terminal value of its net position, the net cash flow of the trader from its trades over the course of the simulation, and the private benefits it receives from its net position. Let $\mathbf{q_T}$ denote the terminal net position of ZI_i , let $\mathbf{r_T}$ denote the terminal fundamental value of the security, and let **profits** be the net cash flow from trades. Then, at the end of the simulation, we calculate the following for the trader's surplus:

```

1     surplus = profits
2     surplus += q_T * r_T
3     private_value = 0
4     if q_T > 0:
5         for q in [1, q_T]:
6             private_value += thetas[q]
7     elif q_T < 0:
8         for q in [q_T+1, 0]:
9             private_value -= thetas[q]
10
11     surplus += private_value

```

B.3.12 LATENCY ARBITRAGE (LA)

The latency arbitrageur (LA) in the two-market model operates as follows. At initialization, it subscribes to the BBO updates from both exchanges. During the simulation, whenever the LA agent's `update_bbo()` function is called by an exchange, it updates its internal BBO for that exchange and then evaluates whether there is an arbitrage opportunity:

```
1  if executing_strategy:
2      return
3
4  max_bid = None
5  min_ask = None
6  for exchange, BBO in BB0s:
7      if max_bid == None or
8          (BBO.bid != None and BBO.bid > max_bid):
9          max_bid = BBO.bid.
10         bid_ex = exchange
11      if min_ask == None or
12          (BBO.ask != None and BBO.ask < min_ask):
13          min_ask = BBO.ask
14          ask_ex = exchange
15  if max_bid != None and min_ask != None and
16      max_bid > (1 + self.alpha) * min_ask:
17      executing_strategy = True
18      midpoint = (max_bid + min_ask) / 2
19      submit_order(ask_ex, 'buy', price=floor(midpoint), q=1)
20      submit_order(bid_ex, 'sell', price=ceil(midpoint), q=1)
21      executing_strategy = False
```

Note that we implement a lock to ensure the LA strategy can complete in full before it submits new orders. Otherwise it might receive a quote update from the exchange and think there's a new arbitrage opportunity, kicking off more orders prior to the other leg of the original strategy finishing. When its orders result in trades (which they are guaranteed to do, since it is not subject to any latency) the LA agent adds the cash flow to its total profits. For a given arbitrage opportunity it acts on, it

achieves a profit of $BID^* - ASK^*$ from the resulting trades. The surplus for the LA trader is just its profits (i.e. net cash flow).

B.3.13 METRIC CALCULATION

After the scheduler finishes its scheduled events, we calculate the following metrics on the events from the simulation: the total ZI surplus, LA profits, mean execution time, median spread, and the number of transactions.

Spread

WW describe the spread as the amount by which the best ask (ASK) exceeds the best bid (BID). They report the median spreads for the BBO and the NBBO. They do not address how to handle a locked or crossed market when calculating spread. We assume we should remove observations where the BID or ASK are empty or where the market was crossed (i.e. require $ASK \geq BID$)³. WW also do not clarify whether the median is calculated on the spread from each quote update or if the spread should be calculated once per time step. We assume it should be calculated for each update (i.e. tick time, rather than clock time). The NBBO median spread is then taken as the median value from the spreads calculated for each NBBO update where the BID and ASK are both non-empty and $ASK \geq BID$. One other important note on the NBBO spread is it is not clear whether NBBO updates scheduled for $t > T$ should be included in the calculation (which can happen when latency $\delta > 0$). We assume no events are executed for $t > T$, and thus these NBBO updates are ignored in our

³*MarketSim* (code) calculates spread as infinity when BID or ASK are null or when $ASK < BID$.

implementation.

When there is a single exchange in the market, the BBO spread is the median of all spreads on BBO updates where the BID and ASK are both non-empty⁴. For configurations with two exchanges, the median is calculated for each exchange BBO and then the average is taken between the two.

Execution time

WW say, “We measure execution time by the interval between order submission and transaction for orders that eventually trade” [189, p. 79]. We assume the execution time should be calculated for both legs of the trade, therefore meaning at least one leg will have an execution time of zero for each trade. Thus, the mean execution time is the average time to execution (i.e. execution time minus submission time) for all orders that resulted in trades.

Transactions

The logic to calculate the number of transactions is inferred from Fig. 11 in WW [189, p. 90] and the behavior of the model. Namely, transactions are the total *orders* that resulted in trades throughout the simulation, separated by the type of trader whose order it was. Thus, a trade between two ZI traders results in two ZI transactions, while a trade between a ZI trader and the LA agent is logged as one ZI transaction and one LA transaction.

⁴Note that $ASK > BID$ at the exchange-level by definition of the CDA.

Surplus

Finally, surplus is calculated as follows. The total ZI surplus is calculated by summing the terminal **surplus** values for each ZI_i in the simulation (see the ZI submodel section for how the **surplus** is calculated for each ZI trader). The LA trader only has the profit it achieved from the net cash flow of its trades. We take the LA agent's terminal profits as its total surplus.

APPENDIX C

SUPPLEMENTARY MATERIAL FOR CHAPTER 4

C.1 ODD PROTOCOL

Below, we give the model description, which follows the ODD (Overview, Design concepts, Details) protocol [90, 92] for our base extension of Wah and Wellman’s 2016 model (WW) [189]. We follow as closely as possible the model as specified by the text of WW, except we extend the model in the following ways. We move the NBBO routing inside the exchange. We allow for variable order size, updating the exchange and latency arbitrageur logic accordingly. We add an Observer entity to track the direct best bid and offer (DBBO) prices in real time. We also introduce additional trading strategies for the background agents, based on Vyetrenko et al. [187] and the ABIDES platform [36]

C.1.1 OVERVIEW

Purpose

The model examines the effects of fragmentation, latency, and latency arbitrage in a financial market. The model features a single stock traded by background trader agents in a market with one or more stock exchange. The model can be run for a consolidated market with a single exchange, or with the incremental elements of an additional exchange, communication latency, and a latency arbitrageur. This allows for the incremental effects of these features on aggregate behavior to be examined.

Entities	Scheduler: schedules and kicks off events in the simulation Exchange: market operating a continuous double auction (CDA) Security: fungible asset traded on each exchange Security Information Processor (SIP): data consolidator for the market; maintains and publishes the National Best Bid and Offer (NBBO) Observer: data consolidator that tracks the direct best bid and offer (DBBO) in the market, with no communication latency Traders: Zero Intelligence (ZI) background traders Latency Arbitrageur (LA) Noise background traders Value background traders Market Maker (MM) background traders Momentum background traders
-----------------	--

Table C.1: Entities in the model ODD protocol.

State variables	Scheduler
	Current time t
	List of scheduled events (background trader activations, SIP updates)
	Exchange
	Limit order book (LOB): all resting orders with prices, quantities, and timestamps
	Best bid and offer (BBO): highest buy price and lowest sell price on the exchange
	National best bid and offer (NBBO): NBBO published by the SIP
	BBO subscribers
	SIP
	Exchange BBO(s): Current BBO feeds from the exchange(s)
	National Best Bid and Offer (NBBO)
	NBBO subscribers
	Observer
	Exchange BBO(s): Current BBO feeds from the exchange(s)
	Direct Best Bid and Offer (DBBO)
	Trades
	Security
	Fundamental value r_t

Table C.2: State variables for the non-trader entities in the model ODD protocol.

State variables	ZI trader
	Valuation vector Θ_i : marginal private benefits for buying and selling the asset
	Strategy parameters $\{R_{\min}, R_{\max}, \eta\}$
	Local exchange
	Local exchange BBO
	Current holdings: quantity q_i the trader is long or short
	Outstanding orders
	Profits from trading
	LA trader (if present)
	Exchange BBOs
	executing_strategy lock flag
	Profits from trading

Table C.3: State variables for the WW trader classes in the model ODD protocol.

State variables	Noise trader
	Order size
	Local exchange
	Subscribed feed (NBBO or local exchange BBO)
	Current holdings: quantity q_i the trader is long or short
	Outstanding orders
	Profits from trading
	Value trader
	Order size
	Fundamental estimate $\hat{r}_t$
	Variance estimate σ_t^2
	Previous wake time
	Local exchange
	Subscribed feed (NBBO or local exchange BBO)
	Current holdings: quantity q_i the trader is long or short
	Outstanding orders
	Profits from trading
	MM trader
	Order size
	Last observed trade
	Default spread
	Levels to place orders at
	Local exchange
	Subscribed feed (NBBO or local exchange BBO)
	Current holdings: quantity q_i the trader is long or short
	Outstanding orders
	Profits from trading
	Momentum trader
	Order size
	Midpoint history
	Local exchange
	Subscribed feed (NBBO or local exchange BBO)
	Current holdings: quantity q_i the trader is long or short
	Outstanding orders
	Profits from trading

Table C.4: State variables for the Get Real [187] trader classes in the model ODD protocol.

Scales	T : Simulation length $N_{exch} \in \{1, 2\}$: Number of exchanges $\delta \geq 0$: SIP latency N_{ZI} : Number of ZI traders $N_{LA} \in \{0, 1\}$: Presence or absence of the LA trader λ : ZI activation rate $q_{\max}$: maximum shares ZI trader can be long or short the asset
---------------	--

Table C.5: Scales in the model ODD protocol.

Process overview and scheduling

Time is discrete, with steps $t \in 1, \dots, T$. A scheduler maintains a queue of exogenous events, ordered by time and then by insertion order (first-in, first-out). At the scheduler level, all trader activation events and all delayed SIP update events are queued. Delayed SIP events include both NBBO updates and trade-feed updates when latency is positive. Exchange matching, BBO publication, trade confirmations, routing, and LA reactions occur immediately as consequences of scheduled events.

C.1.2 DESIGN CONCEPTS

Basic principles

Underlying the model is the concept that fragmentation in a market leads to communication latency. This latency affects different classes of traders unequally. Background investors are assumed to trade based on some understanding of the fundamental value of the security and additional private factors. These investors are less invested in speed technologies than the HFT traders and therefore have some exposure to stale information in the market relative to HFT traders. This is represented in the

model through a simplified latency arbitrage scenario, in which the SIP connecting the markets represents possibly stale information (when latency $\delta > 0$) relative to the direct access the HFT arbitrage agent has to prices at each exchange. These assumptions (fragmentation, speed/information differentials, and high-level trading strategies) determine the structure and individual behaviors in the model. The model provides insight into the impact these features have on the system overall through various market-level performance characteristics.

Emergence

The key results examined in the model are statistical properties of the price return series and market quality metrics related to trading activity on the exchanges. These features are emergent from the interactions of traders adding orders in the simulation.

Adaptation

No details/parameters of the ZI or LA strategies change over the course of the simulation in order to improve profitability.

Objectives

Traders aim to maximize their surplus (private value plus profit), but they are not strategic beyond what they are hard-coded to do. The LA agent submits orders immediately at prices ensured to execute and achieve immediate profit

Learning

The agents do not learn or adapt their traits over the course of the simulation.

Prediction

No prediction is done by the agents.

Sensing

Traders use data feeds in the market to sense the prices available. This determines whether to act in the LA agent's case, and it determines the order routing decision for the ZI agents.

Interaction

Traders interact through the matching of their orders in the LOBs at the exchanges. The resting orders at each exchange inform the routing decisions of the subsequent traders placing orders. Trader actions on the LOBs are communicated via BBO and NBBO updates, which are received as they are updated. These updates occur instantly for the BBOs and with latency δ for the NBBO after events in the model.

Stochasticity

The fundamental price of the security is a mean-reverting stochastic process. ZI trader's internal valuations and order prices are assumed to be stochastic perturbations on top of the security's fundamental value. Exact ZI agent activation times are also stochastic, with individual agent interarrival times drawn from an exponential distribution.

Collectives

Equal sized groups of ZI traders are associated with each of the exchanges. This association determines which exchange’s direct BBO feed the trader receives and determines the trader’s default routing decision. These features in turn factor into the trader’s specific actions. Aside from this difference between groups of ZI traders, there are no collectives in the model.

Observation

The price quotes and trades in the market are logged by the exchanges and published to be logged by the Observer DBBO and the SIP NBBO. Every time an order is submitted, canceled, or results in a transaction at an exchange, this results in a change to the LOB. As a result, the exchange’s BBO is updated and published to the traders, the SIP, and the Observer. The SIP tracks BBO updates (at a lag δ) in its NBBO. The Observer tracks the BBOs and trades. These observations are used at the end of the simulation to calculate metrics, which are then logged, and the Observer’s trade and direct BBO (DBBO) feeds are also saved.

C.1.3 INITIALIZATION

Before each model run, the security, traders, exchanges, SIP, and Observer are instantiated. First, we create the exchanges and the SIP. The SIP’s NBBO and exchange BBO(s) are initialized to have null values as their best bid and best offer. The SIP subscribes to the direct BBO feed from each exchange. The Observer similarly initializes its DBBO to have null values for the best bid and best offer and subscribes

to the direct BBO feed from each exchange. The security traded in the market is also initialized. This initialization logic is detailed in Section C.1.11, with different initialization steps taken depending on what type of exogenous fundamental value logic the traders rely on. Finally, the Observer is initialized. It subscribes to the BBO and trade feeds from each exchange.

Next, the background traders are created. The order these traders are created in is the following, skipping any traders which have zero count specified in the agent configurations:

1. ZI
2. Noise
3. Value
4. MM
5. Momentum
6. LA

The logic for initializing each of these agent types is detailed in their respective sections below. Once these initialization steps have been completed, the simulation is kicked off.

C.1.4 INPUT DATA

The model does not use input data aside from the ZI strategy mixture probabilities (discussed above) and the parameters, which are specified in the next subsection.

C.1.5 PARAMETERS

The three environments specified by WW are given in Table C.6. The parameter N_{ZI} is the number of ZI traders, λ is the activation rate of ZI traders, κ is the strength of mean reversion of the security’s fundamental value, T is the simulation length, and Δ_δ is the granularity of latency steps across experiments for a given environment. The parameters that are fixed across environments are shown in Table C.7.

Environment	N_{ZI}	λ	κ	T	Δ_δ
1	24	0.05	0.05	15,000	100
2	238	0.005	0.02	10,000	10
3	58	0.005	0.02	5,000	10

Table C.6: Parameter settings for the three market environments from WW [189, p. 80]. Please note, the λ parameter for Environments 1 and 2 has an apparent typo ($\lambda = 0.0005$) in WW’s Table 3 [189, p. 80]; the 0.005 value is specified in the text [189, p. 80] and in Wah’s dissertation [188].

Parameter	Default	Description
r_{mean}	100,000	Mean fundamental value of security.
σ_{shock}^2	5,000,000	Variance of the random shock to the security’s fundamental value.
σ_{PV}^2	5,000,000	Variance of ZI private valuations; used to draw private marginal utility of the agent’s net position.
α	0.001	Threshold from which the LA agent determines whether an opportunity is worth pursuing.
q_{max}	10	Maximum position (long or short) for ZI traders

Table C.7: Default parameters used across WW model configurations. These were specified by Wah and Wellman [189, p. 80] except for α . The α value is based on the 2013 paper [190] and Wah’s dissertation [188], and it is further confirmed by looking at the default value α in MarketSim.

ID	R_{min}	R_{max}	η
ZI_1	0	125	1
ZI_2	0	250	1
ZI_3	0	500	1
ZI_4	250	500	1
ZI_5	0	1000	1
ZI_6	500	1000	0.4
ZI_7	500	1000	1
ZI_8	0	1500	0.6
ZI_9	1000	2000	0.4
ZI_{10}	0	2500	0.4
ZI_{11}	0	2500	1

Table C.8: ZI strategy combinations, taken from Wah and Wellman [189, p. 80].

Env	Model	Latency	ZI_1	ZI_2	ZI_3	ZI_4	ZI_5	ZI_6	ZI_7	ZI_8	ZI_9	ZI_{10}	ZI_{11}
1	CDA	-	0	0	0	0	0	0	0	0	0.507	0.493	0
1	2M no LA	0	0	0	0	0	0	0	0	0	0	1.0	0
1	2M no LA	100	0	0	0	0	0	0.602	0	0	0.239	0.159	0
1	2M (LA)	100	0	0	0	0	0	0.237	0	0	0.537	0.226	0
1	2M no LA	200	0	0	0	0	0	0.381	0	0	0.338	0.281	0
1	2M (LA)	200	0	0	0	0	0	0	0	0	0.679	0.321	0
1	2M no LA	300	0	0	0	0	0	0.692	0	0	0.036	0.272	0
1	2M (LA)	300	0	0	0	0	0	0	0	0	0.655	0.345	0
1	2M no LA	400	0	0	0	0	0	0	0	0	0.595	0.405	0
1	2M (LA)	400	0	0	0	0	0	0.47	0	0	0.258	0.272	0
1	2M no LA	600	0	0	0	0	0	0.81	0	0	0	0.19	0
1	2M (LA)	600	0	0	0	0	0	0	0.029	0	0	0.971	0
1	2M no LA	700	0	0	0	0	0	0.739	0	0	0	0.261	0
1	2M (LA)	700	0	0	0	0	0	0.006	0	0	0.826	0.168	0
1	2M no LA	900	0	0	0	0	0	0	0	0	0	1.0	0
1	2M (LA)	900	0	0	0	0	0	0.131	0	0	0	0.869	0
2	CDA	-	0	0	0	0	0	0	0	0	0	0.659	0.341
2	2M no LA	0	0.146	0	0	0	0	0	0	0	0	0.854	0
2	2M no LA	50	0	0.162	0	0	0	0	0	0	0	0.832	0
2	2M (LA)	50	0	0	0.188	0	0	0	0	0	0	0.812	0
2	2M no LA	100	0.051	0	0	0	0	0	0	0	0	0.76	0.189
2	2M (LA)	100	0	0	0	0	0	0	0	0	0.233	0.767	0
3	CDA	-	0	0	0	0	0	0	0	0	0.248	0.752	0
3	2M no LA	0	0	0	0.017	0	0	0	0	0	0.004	0.979	0
3	2M no LA	25	0	0	0	0	0	0	0	0	0	0.854	0.146
3	2M (LA)	25	0	0	0	0	0	0	0	0	0.21	0.79	0
3	2M no LA	50	0	0	0	0	0	0	0	0	0	0.948	0.052
3	2M (LA)	50	0	0	0	0	0	0	0	0.065	0.043	0.892	0
3	2M no LA	75	0	0	0	0	0	0	0	0	0	0.823	0.177
3	2M (LA)	75	0	0	0	0	0	0	0	0	0.142	0.858	0
3	2M no LA	100	0	0	0	0	0	0	0	0	0	0.839	0.161
3	2M (LA)	100	0	0	0.015	0	0	0	0	0	0.231	0.754	0

Table C.9: ZIRP surplus-maximizing equilibrium mixtures for each experiment (environment+configuration+latency). ZIRP strategy settings (ZI_i) are as specified in Table C.8.

C.1.6 SIMULATION DRIVER

The simulation driver initializes the simulation by creating the scheduler, exchanges, traders, security, and SIP. It kicks off the initialization functions for each of these according to the logic detailed in the Initialization section. This results in a population of exchanges, traders, a security with a time-dependent fundamental value, and data feeds in the market. The simulation driver then kicks off the schedule to run through events. The scheduler adds events along the way as messages are sent by agents and as ZI agents reschedule themselves for activation. When the scheduler finishes all events, the simulation driver kicks off the functions to save the data from the simulation for analysis.

C.1.7 SCHEDULER

The scheduler maintains the evolving schedule of events, kicking them off based on their time. WW say, “[E]vents are maintained in a queue ordered by time of occurrence” [189, p. 78], and Wah’s dissertation says, “Multiple events may be scheduled for the same time step, in which case they are executed deterministically in the order in which they are enqueued” [188, pp. 16–17]. We therefore assume events are prioritized first by time, then by order of scheduling (i.e., first-in-first-out for each time step).

An event that has sub-events will kick off those related actions directly instead of going through the scheduler. When the simulation is started, the scheduler kicks off each event based on its time and the order in which it was enqueued for the current time step. Once there are no more events at the current time step, the scheduler skips

ahead to the next time step which has a scheduled event. No events are scheduled for $t > T$.

C.1.8 EXCHANGES

Each exchange in the model operates a continuous double auction (CDA)¹ for the asset. A CDA is a two-sided market where agents submit orders to buy and sell the asset. The market is ‘continuous’ in the sense that orders and trades occur as soon as they are received and processed by the exchange’s matching engine. A limit order to buy (sell) the stock specifies a maximum (minimum) amount the agent is willing to trade at and a quantity to buy (sell). Order specify a positive integer quantity of the stock to buy or sell. Each exchange maintains a limit order book (LOB), constructed with the following elements:

1. Orders: map from order IDs to their respective orders
2. Bids: map from price to the queue of buy order IDs at that price
3. Asks: map from price to the queue of sell order IDs at that price

The exchange publishes a BBO to its subscribers, a feed containing the best bid and offer prices in the LOB at that moment. LOBs are initialized with no bids or asks at the start. The BBO has null prices for its best available bid and ask whenever their respective side of the LOB is empty (including at initialization).

¹Wah and Wellman [190, 189] also examined an alternative clearing mechanism, periodic call auctions, as a possible remedy to latency arbitrage. All exchanges in the U.S. stock market currently use CDAs, however, and we leave the periodic call auction models out of scope for now.

Order submission

When an exchange receives an add message for order O_i with limit price p , it first checks whether the order should be rerouted to another exchange based on the NBBO. If there is only one exchange, this step can be skipped. Assume `order.price` is set to p and `order.exchange` is set to the current exchange. Also assume **BBO** is the current exchange's BBO and **NBBO** is the current NBBO received from the SIP. The logic checks whether the NBBO indicates that the order would immediately execute at a price better than the exchange's BBO. If so, the order is rerouted to the exchange holding the appropriate side of the NBBO.

```
1  submit_nms_order(order):
2      NBBO_price_better = False
3      will_transact = False
4      if order.type == "BUY":
5          if NBBO.ask.price != Null:
6              if (BBO.ask == Null) or (NBBO.ask.price < BBO.ask):
7                  NBBO_price_better = True
8                  if order.price >= NBBO.ask.price:
9                      will_transact = True
10                 alt_ex = NBBO.ask.exchange
11 elif order.type == "SELL":
12     if NBBO.bid.price != Null:
13         if (BBO.bid == Null) or (NBBO.bid.price > BBO.bid):
14             NBBO_price_better = True
15             if order.price <= NBBO.bid.price:
16                 will_transact = True
17             alt_ex = NBBO.bid.exchange
18
19 if NBBO_price_better and will_transact:
20     \\ Check whether the NBBO exchange is the current exchange
21     if alt_ex != self:
22         order.exchange = alt_ex
23         order.rerouted_flag=True
24
25     order.exchange.submit_order(order)
```

Adding an order

Say that an order has been received by the exchange's `submit_order()` function, and let the order have price p and quantity s . The exchange first checks whether the added order can be matched against existing orders in the book. More explicitly, if incoming order O_i is a buy order, the exchange checks whether there is a resting sell order O_j with price $q \leq p$. Similarly, if O_i is a sell order, the exchange checks whether there is a resting buy order O_j with price $q \geq p$. In either of these cases, the orders are matched at price q , resulting in a transaction. If multiple resting orders exist on the LOB that could match with the incoming order, the oldest order is given priority (i.e., price-time priority). Say O_j has quantity s' . The transaction is then for $\min(s, s')$. The trade is processed using the logic detailed below (Section C.1.8). The exchange also publishes the trade to its trade feed subscribers. If O_j is filled by the transaction (i.e., the transaction is for s' shares), it is removed from the LOB; otherwise, it remains on the LOB with the remaining quantity $s' - s$. If O_i instead did *not* immediately match, it is added to the LOB. The exchange then updates its BBO (Section C.1.8). Finally, if O_i matched but was not filled by the transaction (i.e., the transaction is for $s' < s$ shares), the remaining order with quantity $s - s'$ shares is reprocessed, going through the `submit_nms_order()` logic again unless the order was flagged by the trader to not be rerouted.

Canceling an order

A trader can request to cancel its own resting order on the LOB. Upon receiving a cancel request, the exchange removes the corresponding order. The exchange publishes a BBO update to its subscribers. As there is no communication latency between

exchange and trader, no confirmation is sent to the trader—it just removes the order from its internal records once the exchange process completes.

Best Bid and Offer (BBO) updates

The exchange best bid and offer (BBO) tracks the best bid and best ask prices available on the exchange’s LOB at any point in time. Whenever a change occurs to the exchange’s LOB due to a cancel, add, or trade, the exchange updates its BBO. It also calculates what quantity of shares it has available at the best bid and what quantity of shares it has available at the best ask. When the BBO is updated, the exchange publishes a BBO update to its subscribers with the following: the current time, the bid price, the bid quantity available, the ask price, and the ask quantity available. This is published regardless of whether the best bid or offer prices were changed by the LOB update. The exchange loops over its subscribers in the order they subscribed to the exchange’s BBO feed, directly calling the subscriber’s `update_bbo()` function.

Trade processing

The exchange sends receipts of the trade to the respective traders associated with each order. After sending the receipts to the traders involved in the trade, the exchange also publishes the trade to any subscribers to its trade feed.

C.1.9 THE SIP

The role of the SIP in the simulation is to consolidate information from across the fragmented market and publish the national best bid and offer (NBBO). This process

is kicked off whenever the SIP receives a BBO update from an exchange. We assume that when latency δ is zero the SIP should update its NBBO (and subscribers) without going through the scheduler. In our implementation, this means when the SIP receives an `update_bbo()` call from an exchange and $\delta = 0$, the SIP directly calls its own `update_nbbo()` function. This prevents agents acting on the same time step from having stale information due to the SIP being queued behind their already-scheduled action. When $\delta > 0$, we assume an `update_nbbo()` call is scheduled at time step $t + \delta$ using the Scheduler, with no priority preference. In other words, if a background agent is already scheduled for that time step, the NBBO update will occur after that agent makes its trading action (and any resultant actions).

When `update_nbbo()` is called, the NBBO is calculated by finding the best BID (ASK) at one exchange and then seeing whether the next exchange has a higher (lower) BID (ASK). If neither exchange has a BID (ASK) order, the NBBO BID (ASK) is empty. WW do not specify how ties for the best BID or ASK should be handled. We arbitrarily award ties to the first exchange in the list. NBBO updates specify the time t of the update, the best BID price and exchange, and the best ASK price and exchange. Once calculated, NBBO updates are published to the subscribers of the SIP in the order in which they subscribed. This is done by looping through the subscribers and directly calling each subscriber's `update_nbbo` function.

Similarly, the SIP is also subscribed to the trade feeds from the exchanges. When `update_trades()` is called, the SIP either directly calls its own `update_sip_trades()` function (if $\delta = 0$), or a call to its `update_sip_trades()` function is scheduled to occur at $t + \delta$. The SIP trade update calls the `update_trades()` function of any subscribers to the trade feed, passing along the received trade.

C.1.10 OBSERVER

The role of the Observer in the simulation is to consolidate information from across the fragmented market, similar to the SIP, except without latency. The Observer maintains a direct best bid and offer (DBBO) series, which is saved at the end of the simulation in order to conduct statistical analysis on the market's evolving price series. Note that the DBBO and the NBBO will be equivalent, except the DBBO will be δ time steps ahead of the NBBO, and price updates in the last $T - \delta$ time steps of a simulation run will be included in the DBBO but not the NBBO (as no events are allowed past $t = T$).

The Observer subscribes to the BBO and trade updates from each exchange. The Observer updates its DBBO immediately, without going through the scheduler, whenever such it receives a BBO update. The DBBO is calculated by finding the best BID (ASK) at one exchange and then seeing whether the next exchange has a higher (lower) BID (ASK). If neither exchange has a BID (ASK) order, the DBBO BID (ASK) is empty. We arbitrarily award ties to the first exchange in the list. The DBBO updates specify the time t of the update, the best BID price and exchange, and the best ASK price and exchange. When the Observer receives a trade update, it logs the trade and the security's current stored fundamental value. Thus, when the fundamental value is sparse (i.e., for the GetReal value trader configurations), the Observer looks up the most recent sparse fundamental value that the security has logged; it does not advance the fundamental to the current time step. The DBBO and trade log are only internal to the Observer until the end of the simulation, at which point they are saved for later analysis.

C.1.11 SECURITY

The security traded in the market is mostly abstract, except it possesses an evolving ‘common fundamental value’ known or estimated by the traders, depending on the version of the model (WW or GetReal).

WW fundamental value

As noted in the Initialization section, we pre-compute the fundamental value series $\{r_t\}$ for the WW model for efficiency reasons. This series is calculated following the below logic:

```
1  values = {}
2  r = r_mean
3  for i in [1,...,T]:
4      shock = N(0, sigma_shock^2)
5      r = max(0, kappa * r_bar + (1-kappa) * r + shock)
6      fundamental_value[i] = r
```

The mean is r_{mean} , the degree of mean-reversion is κ (**kappa**), and $\mu_t \sim N(0, \sigma_{\text{shock}}^2)$ is a Gaussian random shock drawn independently for each time step.

WW do not specify what r_0 should be, so we assume $r_0 = r_{\text{mean}}$. WW also do not specify how or whether the fundamental value should be made to be an integer (since prices are discrete in the model). We choose not to discretize the fundamental value when it is being calculated. A ZI trader arriving at time t looks up r_t as part of its trading strategy. We assume r_t should be rounded to the nearest integer whenever it is looked up by a trader.

The ZI traders use the estimated terminal value of the security in their trading strategy. We model this logic as a function of the security, which the ZI agents call. The estimate is based on the fundamental value r_t at the time t the trader looks it

up. The estimated terminal value r_t^{est} is then the expected fundamental value at time T , accounting for the mean-reversion process. WW do not specify how r_t^{est} is made to be an integer, nor whether each component to the price should be converted to int or just the final price. We assume that only the estimated terminal value should be rounded after the calculation of r_t^{est} . In other words,

$$r_t^{\text{est}} = \text{round} \left(\left(1 - (1 - \kappa)^{T-t} \right) r_{\text{mean}} + (1 - \kappa)^{T-t} r_t \right). \quad (\text{C.1})$$

GetReal sparse fundamental value

The GetReal version of the model features a much more granular representation of time. Specifically, a full 6.5-hour trading day is simulated, and each timestep t represents a nanosecond. Thus, if we followed the calculation process used by the WW version of the model, the fundamental would need to be calculated for 23400×10^9 values. To avoid this, the ABIDES [36] platform implements sparse fundamental value logic, which is further detailed by Byrd [35]. The sparse fundamental value logic uses a continuous mean-reverting Ornstein-Uhlenbeck (OU) process plus periodic ‘megashocks’ which arrive according to a Poisson process. The shocks are drawn from a bimodal normal distribution with mean zero, modes equally spaced on either side of zero. The OU process only departs from the mean with noisy movements, while the shocks are needed to perturb the series away from the mean, analogous to the discrete mean-reverting random walk from WW.

The specific logic is based on the **SparseMeanRevertingOracle** implemented in ABIDES: <https://github.com/abides-sim/abides/blob/master/util/oracle/SparseMeanRevertingOracle.py>.

C.1.12 ZERO INTELLIGENCE (ZI)

ZI Initialization

Next, we create the ZI traders. Each ZI trader is assigned one of the markets as its local market. In the single-exchange configuration of the market, all ZI traders are assigned to the only exchange. In the two-market configurations, we alternate which exchange an agent is assigned to each time we create an agent. We therefore evenly split the ZI traders between the two exchanges, as there are no environments that feature an odd number of ZI agents. Each ZI agent is assigned trading strategy parameters according to the specified mixture probabilities, specified in Table C.9. We sample ZI trader profiles according to the surplus-maximizing equilibrium mixture probabilities provided by WW for the given experiment (environment + latency + number of LA). The surplus-maximizing equilibrium mixture probabilities are determined by combining the information in Tables 6–11 in WW [189, pp. 83–86]. Let a ‘mixture’ refer to a sample of N_{ZI} profiles according to the mixture probabilities. A mixture is stochastically drawn based on the probabilities for a given run. Each profile from the mixture is assigned to one of the ZI agents. ZI agents subscribe to the BBO feed from their assigned local exchange and to the NBBO feed from the SIP. ZI agents initialize their marginal private benefits vector Θ according to the logic detailed in Eq. C.2

Its holdings, profits, and number of transactions are all initialized to zero. Next, for the given ZI trader ZI_i , we construct Θ_i , the vector of differences in private

benefits of changes to the trader's net position:

$$\Theta_i = \{\theta_i^q\}, \quad (\text{C.2})$$

where $-q_{\max} < q \leq q_{\max}$ and the θ_i 's are sorted such that $\theta^q \geq \theta^{q+1} \forall q$. No discretization logic is specified by WW, nor do they state anywhere that the internal component for the ZI agents should be integers. Hence, we do not round or otherwise discretize the θ values when they are initialized.

Schedule initialization

The initial activations for each of the ZI agents scheduled after all the ZI agents have been instantiated. WW say, "The background traders arrive at the market according to a Poisson process with rate λ " [189, p. 75]. They do not specify how the ZI activation times are made to be discrete. We draw the ZI activation times independently from an exponential distribution ($\lceil \text{Exponential}(1/\lambda) \rceil$). We choose to take the ceiling such that no event will be scheduled at time $t = 0$.

Activation and base trading strategy

Each ZI trader arrives to place an order at least once. We assume the first thing the ZI trader ZI_i does when it acts at time t is schedule its next activation for time $t + \lceil b_i \rceil$, where $b_i \sim \text{Exponential}(\frac{1}{\lambda})$. WW do not specify how ZI activation times should be made discrete. We choose to take the ceiling so the trader does not schedule itself to arrive again on the same time step.

After scheduling its next activation, ZI_i withdraws any outstanding orders from previous turns that have not yet resulted in a trade. It then determines its trading

action for this turn. It randomly chooses (with equal probability) whether to buy or sell on this turn. Traders are not allowed to have a net position of $q_{\max}$ shares long or short of the asset. Thus, if the decision to trade or buy would result in a net position $|q| > q_{\max}$, the trader does not submit an order and exits.

Otherwise, ZI_i looks up the estimated terminal fundamental value r_t^{est} of the security (see the Security submodel section). It then calculates its internal valuation $v_i(t)$ of the asset based on r_t^{est} and its marginal private benefits vector Θ_i :

$$v_i(t) = r_t^{\text{est}} + \begin{cases} \theta_i^{q_i+1} & \text{if buying one unit} \\ \theta_i^{q_i} & \text{if selling one unit} \end{cases}. \quad (\text{C.3})$$

We choose to round $v_i(t)$ to the nearest integer, as WW do not specify how prices are made to be discrete. ZI_i 's order price p is ‘shaded’ some amount away from its private valuation in order to achieve surplus value.

$$p \sim \begin{cases} \max(0, U[v_i(t) - R_{\max}, v_i(t) - R_{\min}]) & \text{if buying} \\ \max(0, U[v_i(t) + R_{\min}, v_i(t) + R_{\max}]) & \text{if selling} \end{cases}, \quad (\text{C.4})$$

where U here denotes an integer uniformly chosen at random within the specified range. Note that the ZI trader only submits orders of size one. The requested surplus for ZI_i is then $s_i(t) = |v_i(t) - p_i(t)|$.

Before submitting its order, ZI_i determines whether there is a greedy opportunity based on p , the trader's η parameter, and the current market information. We use the greedy strategy logic detailed in Chapter 3, which was determined from looking at the *MarketSim* codebase that Wah and Wellman were involved in creating [191].

In the *MarketSim* implementation of the greedy strategy, ZI_i looks at its local exchange's BBO when determining whether there is a greedy opportunity. It updates its order price to $p = v_i(t)$ when there is a greedy opportunity:

```

1     requested_surplus = |self.valuation - price|
2     if side == 'buy':
3         if BBO.ask != None and
4             requested_surplus * eta <= valuation - BBO.ask:
5             price = valuation
6     else: // 'sell'
7         if BBO.bid != None and
8             requested_surplus * eta <= BBO.bid - valuation:
9             price = valuation

```

The order is then submitted to the local exchange, where it may possibly be rerouted based on the NBBO.

Calculation of surplus

At the end of the simulation, WW calculates the total surplus S_i for an agent ZI_i with a terminal net position q_i^T and valuation vector $\Theta_i = \{\theta_j\}$. This is calculated as the sum of the following: the terminal value of its net position, the net cash flow of the trader from its trades over the course of the simulation, and the private benefits it receives from its net position. Let **q_T** denote the terminal net position of ZI_i , let **r_T** denote the terminal fundamental value of the security, and let **profits** be the net cash flow from trades. Then, at the end of the simulation, we calculate the following for the trader's surplus:

```

1     surplus = profits
2     surplus += q_T * r_T
3     private_value = 0
4     if q_T > 0:
5         for q in [1, q_T]:
6             private_value += thetas[q]
7     elif q_T < 0:
8         for q in [q_T+1, 0]:

```

```

9         private_value -= thetas[q]
10
11     surplus += private_value

```

C.1.13 LATENCY ARBITRAGEUR (LA)

LA initialization

At initialization, the LA agent subscribes to both exchanges' BBO feeds.

Base trading strategy

During the simulation, whenever the LA agent's `update_bbo()` function is called by an exchange, it updates its internal BBO for that exchange and then evaluates whether there is an arbitrage opportunity:

```

1     if executing_strategy:
2         return
3
4     executing_strategy = True
5     while True:
6         max_bid = None
7         min_ask = None
8         for exchange, BBO in BB0s:
9             if max_bid == None or
10                (BBO.bid != None and BBO.bid > max_bid):
11                 max_bid = BBO.bid.price
12                 max_bid_size = BBO.bid.quantity
13                 bid_ex = exchange
14             if min_ask == None or
15                (BBO.ask != None and BBO.ask < min_ask):
16                 min_ask = BBO.ask
17                 min_ask_size = BBO.ask.quantity
18                 ask_ex = exchange
19
20         if max_bid == None or min_ask == None:
21             break
22         if max_bid <= (1 + self.alpha) * min_ask:
23             break
24

```

```

25     order_size = min(max_bid_size, min_ask_size)
26     midpoint = (max_bid + min_ask) / 2
27     submit_order(ask_ex, 'buy', price=floor(midpoint), q=order_size)
28     submit_order(bid_ex, 'sell', price=ceil(midpoint), q=order_size)
29
30     executing_strategy = False

```

Note that we implement a lock to ensure the LA strategy can complete in full before it submits new orders. Otherwise it might receive a quote update from the exchange and think there's a new arbitrage opportunity, kicking off more orders prior to the other leg of the original strategy finishing. It also keeps reevaluating whether there is an arbitrage opportunity until it determines there is not. This is necessary due to the fact the quantity available at each price level may differ. The LA trader also skips the exchange's NMS re-routing logic, otherwise one its arbitrage order(s) might be misrouted due to a stale NBBO. When the orders result in trades (which they are guaranteed to do, since it is not subject to any latency), the LA agent adds the cash flow to its total profits. For a given arbitrage opportunity it acts on, it achieves a profit of $BID^* - ASK^*$ from the resulting trades. The surplus for the LA trader is just its profits (i.e., net cash flow).

C.1.14 NOISE AGENTS

We implement the 'Noise Agent' strategy outlined by Vyetrenko et al. [187] in the 'GetReal' paper. We supplement the limited details from the paper with implementation details publicly available in the ABIDES platform [36].

Noise initialization

The Noise agents each trade once per day. Their activation times are sampled from a U -quadratic distribution over the interval $[t_{\text{open}}, t_{\text{close}}]$, where t_{open} and t_{close} are the open and close times of the trading day. The specific activation time logic is implemented as follows, based on the ABIDES code. First, a random number $u \in [0, 1)$ is uniformly drawn. The U -quadratic distribution is parameterized by a and b . Let $\beta = \frac{b+a}{2}$ and $\alpha = \frac{12}{(b-a)^3}$. The activation time scalar s is then calculated by passing u into the inverse function of the U -quadratic distribution CDF:

$$\frac{\alpha}{3} \left((x - \beta)^3 + (\beta - a)^3 \right)$$

Thus,

$$s = \beta + \sqrt[3]{\frac{3y}{\alpha} - (\beta - a)^3}.$$

Finally, the activation time is:

$$w = s * (t_{\text{close}} - t_{\text{open}}).$$

Each Noise agent draws a random integer order size uniformly from the range $[20, 50)$ at initialization, which it then uses as its order size when executing its strategy. Each Noise agent is arbitrarily assigned a local exchange. The first noise agent is assigned its exchange at random, and the rest are subscribed to the next exchange, following a round-robin strategy. Under the NBBO information setting, the Noise agent will subscribe to the SIP NBBO. Under the BBO information setting, it will subscribe to the BBO feed from its local exchange. In the below, we will refer to the

(N)BBO, which should be assumed to be the NBBO under the NBBO information setting and the BBO under the BBO information setting.

Activation

Vyetrenko et al. describe the noise agents as acting randomly and exclusively submitting market orders. The noise agents randomly choose to buy or sell, with equal likelihood of either choice. The ABIDES platform implements a consolidated market, but it allows for communication latency between agents in the model (including the exchange agent and traders). The publicly available code implementing the noise agent strategy actually submits limit orders equal to the opposite side of the market BBO that the agent has observed (with latency). If the opposite side of the market BBO is empty, it does not submit an order. A submitted order is communicated with some amount of latency to the exchange. Thus, it is possible in that case for the order to no longer be marketable by the time it reaches the exchange, if the BBO has changed.

We choose to implement the version of the strategy from the code. Thus, upon activation, the trader submits a limit order to buy or sell, with price equal to the opposite side of the (N)BBO. If the opposite side of the (N)BBO is empty, the trader does nothing.

C.1.15 VALUE AGENT

Parameter	Default	Description
λ_v	10^{-12}	Activation rate
$\bar{r}_v$	10^5	Asset mean fundamental value
σ_n	10^4	Asset observation noise variance
κ_v	$1.67 * 10^{-15}$	Asset mean reversion parameter
σ_v	10^5	Asset shock variance
percent_aggr	0.1	Percent of time agent will aggress the spread
depth_spread	2	Spread scalar for how deep to post in the book
min_size	20	Minimum order size (inclusive)
max_size	50	Maximum order size (exclusive)

Table C.10: Parameters and defaults for the Value agents. Defaults are based on the ABIDES platform and the `random_fund_diverse` config file (https://github.com/abides-sim/abides/blob/master/config/random_fund_diverse.py).

The Value Agent is also from the ‘GetReal’ paper [187]. We again draw from that paper and the ABIDES platform [36], and specifically the `random_fund_diverse` config file, for implementation details. Vyetrenko et al. describe the Value Agents as being analogous to fundamental traders that, upon activation, buy when the stock is cheap and sell when it is expensive. This assessment is based on a noisy view of an exogenous ‘fundamental’ price series for the stock. Note that the GetReal versions of the model use the sparse fundamental, as discussed in Section C.1.11. The value agents activate according to a Poisson process with rate λ .

Initialization

The initial activation times are drawn uniformly from the interval $[1, 100]$. Each trader initializes a default order size of an integer randomly drawn from the interval

[20, 50). The agent maintains two priors about the fundamental value—the current value and the error—which it initializes as $\hat{r}_t = \bar{r}_v$ and $\sigma_t = 0$, respectively. Similar to the other GetReal strategies, each value agent is arbitrarily assigned a local exchange. The first value agent is assigned its exchange at random, and the rest are subscribed to the next exchange, following a round-robin strategy. Under the NBBO information setting, the Value agent will subscribe to the SIP NBBO. Under the BBO information setting, it will subscribe to the BBO feed from its local exchange. In the below, we will refer to the (N)BBO, which should be assumed to be the NBBO under the NBBO information setting and the BBO under the BBO information setting.

Activation

Upon activation at time t , the trader first reschedules its next activation for $t + \text{round}(a)$, where $a \sim \text{Exponential}(1/\lambda_v)$. Next, the trader cancels any outstanding orders before placing any new orders.

Let r_t be the fundamental value of the security at time t . The value agent maintains two priors, estimates $\hat{r}_t$ and σ_t of the fundamental value and error. The trader observes a noisy estimate of r_t when it looks up the value at t . Let this noisy observation be O_t :

$$O_t \sim N(r_t, \sigma_n).$$

Then O_t is rounded to the nearest integer.

Let Δ_t be the number of time steps since the trader last acted, or the number of time steps since the start of the simulation if this is the trader's first activation. The trader estimates the current fundamental based on its previous estimate $\hat{r}_t$ and the long-term mean $\bar{r}$. If this is the trader's first action, its previous $\hat{r}_t$ is the starting

value of the security. The updated estimate is:

$$\hat{r}'_t = (1 - (1 - \kappa)^{\Delta_t}) * \bar{r} + (1 - \kappa)^{\Delta_t} \hat{r}_t$$

The trader also updates σ_t :

$$\sigma'_t = (1 - \kappa)^{2\Delta_t} \sigma_t + \left(1 - \frac{(1 - \kappa)^{2\Delta_t}}{1 - (1 - \kappa)^2}\right) \sigma_v^2,$$

where σ_v^2 is the estimated long run variance. Then, $\hat{r}_t$ and σ_t are updated as follows:

$$\hat{r}_t = \frac{\sigma_n}{\sigma_n + \sigma'_t} \hat{r}'_t + \frac{\sigma'_t}{\sigma_n + \sigma'_t} O_t$$

$$\sigma_t = \frac{\sigma_n \sigma_t}{\sigma_n + \sigma_t}.$$

Then, given the estimate of the current value of the asset, the trader estimates the value at the end of the trading day:

$$r_T = \text{round} \left((1 - (1 - \kappa)^{T-t}) \bar{r} + ((1 - \kappa)^{T-t} \hat{r}_t) \right).$$

If either side of the (N)BBO is empty, the agent sets its price equal to r_T and randomly chooses to buy or sell, with equal probability. Otherwise, let m and s be the midpoint and spread of the (N)BBO, respectively. If $r_T < m$, the trader will sell the security (**side='sell'**); otherwise, it will buy (**side='buy'**). With some probability p_{aggr} , the agent will submit an 'aggressive' order, wherein it does not adjust its order and instead sets its price equal to the ask if buying or the bid if

selling. Otherwise,

$$p = \text{ask} - q$$

if buying or

$$p = \text{bid} + q$$

if selling, where q is a random integer in the range $[0, \text{depth_spread} * s)$ and **bid** and **ask** are the (N)BBO bid and ask, respectively. Note that this differs slightly from the prose [187], in that the limit order could be equal to the bid or ask. Not to mention the aggressive order probability, which is also not mentioned. The trader places a limit order for quantity **size** to its local exchange.

C.1.16 MARKET MAKER AGENT

Parameter	Default	Description
num_ticks	5	Number of price levels to place orders at on each side of the book
wake_up_freq	60×10^9 (i.e., <i>1min</i>)	How long between activations
mm_size	100	Total order size for price level
last_spread	10	Default spread value

*Table C.11: Parameters and defaults for the Market Maker agent. Defaults are based on the ABIDES platform and the **random_fund_diverse** config file (https://github.com/abides-sim/abides/blob/master/config/random_fund_diverse.py).*

The Market Maker (MM) agent is also based on the strategy outlined by Vyetenko et al. [187] and implemented in the ABIDES platform [36]. The MM agent provides liquidity on both sides of a LOB (buy and sell limit orders). The MM agent activates at a regular interval (e.g., every minute).

Initialization

We choose to have the initial activation time drawn for the MM agent drawn from an exponential distribution with scale `wake_up_freq`, taking the ceiling to make activation times integers. The MM agent initializes its internal values according to its parameters (Table C.11). The MM's order sizes are fixed at `mm_size/2`. Similar to the other GetReal strategies, the MM agent is arbitrarily assigned a local exchange. The first MM agent is assigned its exchange at random, and any others are subscribed to the next exchange, following a round-robin strategy. Under the NBBO information setting, the MM agent will subscribe to the SIP NBBO. Under the BBO information setting, it will subscribe to the BBO feed from its local exchange. In the below, we will refer to the (N)BBO, which should be assumed to be the NBBO under the NBBO information setting and the BBO under the BBO information setting.

Activation

Upon activation at time t , the MM agent first schedules its next arrival at $t + \text{wake_up_freq}$. It then cancels any of its outstanding orders. The MM agent next determines the prices it will submit orders at. The MM agent executes a basic market making strategy. Specifically, it calculates the following:

```
1    // First, it sets its default midpoint value to be
2    //    the most recent trade price
3    // Assume it has access to a list of all trades published
4    //    by exchange/SIP data feed so far
5    mid = trades[-1]
6
7    // Next, it looks up the most recent (N)BBO
8    bid = (n)bbo['bid']['price']
9    ask = (n)bbo['ask']['price']
10
11    // So long as both sides of the (N)BBO are populated,
12    //    it calculates the midpoint and the spread, rounding
```

```

13     if bid and ask:
14         mid = floor((ask + bid) / 2)
15         spread = floor((ask - bid) / 2)
16     else:
17         //Otherwise it uses the default spread
18         spread = last_spread
19
20     num_levels = 2 * num_ticks
21     // It submits buy and sell orders on opposite sides of the book
22     // for num_levels levels
23     for i < num_levels:
24         // Each order size is for quantity mm_size / 2
25
26         // It then places a buy order at the given level
27         buy_price = mid - spread - i
28         buy_order = Order(size=size, side='buy', price=buy_price)
29         ex.submit_nms_order(buy_order)
30
31         // And it places a sell order at the opposite level
32         sell_price = mid + spread + i
33         sell_order = Order(size=size, side='sell', price=sell_price)
34         ex.submit_nms_order(sell_order)

```

C.1.17 MOMENTUM AGENT

Parameter	Default	Description
<code>wake_up_freq</code>	60×10^9 (i.e., 1min)	How long between activations
<code>min_size</code>	1	Minimum order size (inclusive)
<code>max_size</code>	10	Maximum order size (exclusive)

Table C.12: Parameters and defaults for the Momentum agent. Defaults are based on the ABIDES platform and the `random_fund_diverse` config file (https://github.com/abides-sim/abides/blob/master/config/random_fund_diverse.py).

The Momentum agent is based on the strategy outlined by Vyetrenko et al. [187] and implemented in the ABIDES platform [36]. The Momentum agent sells if the mid-price over the past 20 observations is lower than the average mid-price over the

past 50 observations and buys otherwise. Similar to the MM agent, the Momentum agent arrives to the market at a constant rate.

Initialization

We initialize its arrival time from an exponential distribution with scale `wake_up_freq`, taking the ceiling to make activation times integers. It sets its default order size (`size`) to be a random integer drawn from the interval `[min_size, max_size)`. Similar to the other GetReal strategies, each Momentum agent is arbitrarily assigned a local exchange. The first Momentum agent is assigned its exchange at random, and the rest are subscribed to the next exchange, following a round-robin strategy. Under the NBBO information setting, the Momentum agent will subscribe to the SIP NBBO. Under the BBO information setting, it will subscribe to the BBO feed from its local exchange. In the below, we will refer to the (N)BBO, which should be assumed to be the NBBO under the NBBO information setting and the BBO under the BBO information setting. The Momentum agent also maintains a running list of midpoint prices (`mid_list`), which is initialized as an empty array.

Activation

When activated at time t , the Momentum agent first schedules its next activation at $t + \text{wake_up_freq}$. It then cancels any of its orders that are still outstanding.

As noted above, the Momentum agent compares the recent price trend to the longer price trend, submitting a sell order if the recent trend is relatively lower and submitting a buy order otherwise. Note, the Momentum agent logs an observation each time it is activated, not each the NBBO is updated. Thus, it will not place an

order until its 51st activation.

```
1 // First, it looks up the most recent NBBO
2 bid = nbbo['bid']['price']
3 ask = nbbo['ask']['price']
4
5 if bid is not None and ask is not None
6     and bid > 0 and ask > 0:
7     mid_list.append((bid+ask)/2)
8     if len(mid_list) > 50:
9         // Average the 20 and 50 most recent observations, respectively
10        avg_20 = sum(mid_list[-20:]) / 20
11        avg_50 = sum(mid_list[-50:]) / 50
12
13    if avg_20 >= avg_50:
14        // submit buy order at the ask price
15        order = Order(size=size, side='buy', price=ask)
16    else:
17        // submit sell order at the bid price
18        order = Order(size=size, side='sell', price=bid)
19    ex.submit_nms_order(order)
```

C.1.18 METRIC CALCULATION

After the scheduler finishes its scheduled events, data are saved for analysis. The Observer saves the DBBO for the simulation. We also calculate the following metrics, for checking alignment [13] against the canonical WW results: the total ZI surplus, LA profits, mean execution time, median spread, and the number of transactions.

Spread

WW describe the spread as the amount by which the best ask (*ASK*) exceeds the best bid (*BID*). They report the median spreads for the BBO and the NBBO. They do not address how to handle a locked or crossed market when calculating spread. We assume we should remove observations where the BID or ASK are empty or where

the market was crossed (i.e., require $ASK \geq BID$)². WW also do not clarify whether the median is calculated on the spread from each quote update or if the spread should be calculated once per time step. We assume it should be calculated for each update (i.e., tick time, rather than clock time). The NBBO median spread is then taken as the median value from the spreads calculated for each NBBO update where the BID and ASK are both non-empty and $ASK \geq BID$. One other important note on the NBBO spread is it is not clear whether NBBO updates scheduled for $t > T$ should be included in the calculation (which can happen when latency $\delta > 0$). We assume no events are executed for $t > T$, and thus these NBBO updates are ignored in our implementation.

When there is a single exchange in the market, the BBO spread is the median of all spreads on BBO updates where the BID and ASK are both non-empty³. For configurations with two exchanges, the median is calculated for each exchange BBO and then the average is taken between the two.

Execution time

WW say, “We measure execution time by the interval between order submission and transaction for orders that eventually trade” [189, p. 79]. We assume the execution time should be calculated for both legs of the trade, therefore meaning at least one leg will have an execution time of zero for each trade. Thus, the mean execution time is the average time to execution (i.e., execution time minus submission time) for all orders that resulted in trades.

²*MarketSim* (code) calculates spread as infinity when *BID* or *ASK* are null or when $ASK < BID$.

³Note that $ASK > BID$ at the exchange-level by definition of the CDA.

Transactions

The logic to calculate the number of transactions is inferred from Fig. 11 in WW [189, p. 90] and the behavior of the model. Namely, transactions are the total *orders* that resulted in trades throughout the simulation, separated by the type of trader whose order it was. Thus, a trade between two ZI traders results in two ZI transactions, while a trade between a ZI trader and the LA agent is logged as one ZI transaction and one LA transaction.

Surplus

Finally, surplus is calculated as follows. The total ZI surplus is calculated by summing the terminal **surplus** values for each ZI_i in the simulation (see the ZI submodel section for how the **surplus** is calculated for each ZI trader). The LA trader only has the profit it achieved from the net cash flow of its trades. We take the LA agent's terminal profits as its total surplus.

C.2 REPLICATION AND VALIDATION OF WAH AND WELLMAN (2016)

The background traders follow a trading strategy that is inspired by the zero-intelligence (ZI) trading strategy introduced by Gode and Sunder [86]. The traders in Wah and Wellman's 2016 model act in the market multiple times per simulation, versus their original 2013 model where each background trader was activated exactly once. The 2016 strategy also introduces strategic behavior not present in the 2013 model. We

therefore refer to the 2016 strategy as ‘zero intelligence with reentry plus’ (ZIRP) going forward, to differentiate it from the canonical strategy from Gode and Sunder and the basic 2013 strategy from Wah and Wellman [190]. The exact logic of the ZIRP traders is detailed in full in the ODD (Section C.1).

In the original study, Wah and Wellman [189] provide the quantitative values for the mean ZIRP surplus and LA profits for each experiment. All other results from the study are plotted visually. We test the fidelity of our implementation by assessing relational equivalence [13] between our independent replication and the plotted results from Wah and Wellman [189] and our results from Chapter 3. We leave tests of quantitative alignment against the mean surplus and profit values out of scope for this verification exercise. See Chapter 3 for a detailed discussion of testing for quantitative alignment based only on these mean values.

C.2.1 WAH AND WELLMAN’S 2016 MODEL

We begin from our replication of Wah and Wellman’s 2016 model [189] detailed in Chapter 3. As discussed earlier, Wah and Wellman were the first to use an ABM to examine the effects of latency arbitrage in a fragmented market. The model therefore provides an important baseline, and building on it allows us to take a cumulative approach to model and knowledge development [13]. Wah and Wellman introduced their model in a pair of studies from 2013 and 2016. The 2013 version of the model [190] featured simple background trader agents that only entered the market once per simulation. We refer to our implementation of the model as ‘WW’ for simplicity.

The overall setup of the model is described in the main paper (Section 4.3). The model features one or two exchanges, a SIP that publishes the NBBO at some lag δ ,

background investor traders, and an optional latency arbitrageur (LA). In the 2013 version of the model, all other parameters were kept constant, and they reported results from simulating 200 runs of the model per experiment. Wah and Wellman substantially extended the model in 2016. In the updated model, the background agents utilize a more sophisticated trading strategy, have heterogeneous strategy parameter sets, and can participate multiple times per simulation. Wah and Wellman varied the market configurations for similar structural changes as before, but they examined three parameter sets and ran the model for 50,000 runs for each experiment.

We focus on the 2016 version of the model here, due to its more sophisticated experimental design and more detailed reporting of results. We unfortunately do not have the original model code used to produce the published results. We instead utilize an independent replication as our starting point. We stick as closely to the original model for this step. We then test qualitative alignment to and empirically benchmark the base model, detailed below, before ultimately extending the model for the version presented in our main paper.

We will refer to the version of our model replicating Wah and Wellman [189] as ‘WW’ going forward. The background trader agents in WW implement a ‘zero-intelligence’ (ZI) trading strategy inspired by the seminal work of Gode and Sunder [86]. Their specific agent strategy built on the 2013 version of their model, adding logic to allow traders to activate multiple times per simulation and to act strategically to achieve surplus value based on their internal valuation of the stock. We call this agent strategy ‘zero intelligence with reentry plus’ (ZIRP)⁴ to differentiate it from the classic strategy introduced by Gode and Sunder [86]. We describe the high-

⁴The ZIRP name is based on the implementation of this strategy in the *MarketSim* codebase, which Wah and Wellman helped build [191].

level ZIRP strategy here, and a more in-depth description can be found in our ODD (Appendix C.1.12).

ZIRP strategy

At initialization, each ZIRP trader is assigned a local exchange. When there are multiple exchanges in the market, agents are evenly distributed among the exchanges, with exchange assignments following a round-robin strategy. A ZIRP agent will always submit any orders to its local exchange, which then does any rerouting on behalf of the trader based on the NBBO (as discussed in Section 4.3). The agent also has access to its local exchange’s BBO feed but not to any other data feeds in the market⁵. Each trader is also assigned strategy parameters $\{R_{\min}, R_{\max}, \eta\}$. How these strategy parameters are assigned is discussed in further detail in Section C.2.2

Each trader $ZIRP_i$ has a vector Θ_i of the marginal private benefits it receives for changes to its net position:

$$\Theta_i = \left(\theta_i^{-q_{\max}+1}, \dots, \theta_i^0, \theta_i^{+1}, \dots, \theta_i^{q_{\max}} \right),$$

where $q_{\max}$ is the maximum number of shares the trader can be long or short the stock. This vector is constructed from $2q_{\max}$ independent draws from a normal distribution $N(0, \sigma_{PV}^2)$, sorted such that $\theta_i^q \geq \theta_i^{q+1} \forall q$. If the trader has q shares of the stock, its private marginal benefit is θ_i^{q+1} for buying an additional share or θ_i^q for selling a share.

⁵Wah and Wellman [189] describe the ZIRP agents as having access to the NBBO. However, as the local exchange determines the ultimate order routing decision in our implementation, and it does not appear that the trader uses the NBBO elsewhere in its trading strategy. See Chapter 3 for a deeper dive on this nuance.

The ZIRP traders are activated according to a Poisson process with rate λ_{ZIRP} . Interarrival times are independently drawn from an exponential distribution. Specifically, the first thing that a ZIRP trader does when it is activated at time t is to schedule its next activation to occur at $t + \lceil a \rceil$, where

$$a \sim \text{Exponential}(1/\lambda_{\text{ZIRP}}).$$

Initial activation times are drawn from this same distribution (again taking the ceiling, to avoid any activations at time $t = 0$).

Upon activation at time t , and after scheduling its next activation, a trader will cancel any outstanding order it has from its previous turn. It then randomly decides to buy or sell the stock. If this decision would result in the trader being more than q_{max} shares long or short, the trader exits and does not submit an order. Otherwise, it determines the order price according to its private valuation of the stock at that time. This private valuation is based on a shared view of the (exogenous) fundamental value of the stock and the private marginal benefit for trading the stock given how many shares the trader is currently long or short the stock.

The fundamental value of the stock follows a mean-reverting random walk:

$$r_t = \max\{0, \kappa\bar{r} + (1 - \kappa)r_{t-1} + u_t\},$$

where parameter $\kappa \in [0, 1]$ specifies the strength of reversion to the mean $\bar{r}$, the initial value $r_0 = \bar{r}$, and $u_t \sim N(0, \sigma_{\text{shock}}^2)$ is a Gaussian random shock at each time step. The ZIRP traders have a perfect view of the current r_t when they are activated, which they then use to estimate the terminal value r_T . The estimated terminal value

$\hat{r}_t$ is the expected value of r_T given the current value r_t and the underlying stochastic process. This is calculated as follows and rounded to make $\hat{r}_t$ an integer:

$$\hat{r}_t = \text{round} \left(\left(1 - (1 - \kappa)^{T-t} \right) \bar{r} + (1 - \kappa)^{T-t} r_t \right).$$

The private valuation $v_i(t)$ for $ZIRP_i$ at time t is calculated as:

$$v_i(t) = \hat{r}_t + \begin{cases} \theta_i^{q_t+1} & \text{if buying one unit} \\ \theta_i^{q_t} & \text{if selling one unit} \end{cases}.$$

The trader then determines its order price p as:

$$p \sim \begin{cases} \max(0, U[v_i(t) - R_{max}, v_i(t) - R_{min}]) & \text{if buying} \\ \max(0, U[v_i(t) + R_{min}, v_i(t) + R_{max}]) & \text{if selling} \end{cases},$$

where U here denotes an integer uniformly chosen at random within the specified range. The requested surplus for $ZIRP_i$ is then $s_i(t) = |v_i(t) - p_i(t)|$.

Greedy strategy extension The ZIRP strategy introduced by WW features what we refer to as a ‘greedy strategy’ extension. This part of the strategy evaluates whether the agent could immediately achieve a fraction η of its requested surplus $s_i(t)$ given the current price quote at its local exchange. This strategy is implemented as follows. Let **BBO** refer to the most recent BBO received by the trader from its local exchange. Then:

```

1    max_bid = BBO.BID
2    min_ask = BBO.ASK
3    requested_surplus = |v_i(t) - p|
4    if side == 'BUY':
5        if requested_surplus * eta <= v_i(t) - min_ask:
6            p = v_i(t)
7    else: // 'SELL'
8        if requested_surplus * eta <= max_bid - v_i(t):
9            p = v_i(t)

```

Note that in a consolidated market this will always result in the trader’s order executing at the local exchange BBO (either the bid or the ask depending on whether the agent is selling or buying, respectively). However, if the market is fragmented with nonzero latency, it is possible the order will be rerouted instead of being executed at the local exchange, in which case it may execute at a price inferior to the BBO price it was aiming to hit. This is partly due to the fact the agent is setting its order price $p = v_i(t)$ instead of setting $p = \text{min_ask}$ or $p = \text{max_bid}$. This is discussed in further detail in Chapter 3.

C.2.2 WW MODEL EXPERIMENTS

We evaluate the WW model for the experimental settings used by Wah and Wellman in their canonical results [189]. The default parameters used in each experiment are shown in Table C.13. Another set of parameters is varied depending on the ‘environment’, with three environments overall. The environment parameter sets are shown in Table C.14. The parameter N_{ZI} is the number of ZI traders, λ is the activation rate of ZI traders, κ is the strength of mean reversion of the security’s fundamental value, and T is the simulation length. Wah and Wellman varied the SIP latency δ for at least three values for each environment. The specific δ values varied

for each environment, as shown in Table C.14. Within an environment, there are three possible configurations of the market: single exchange (CDA); two exchanges with no LA agent (2M), and two exchanges with LA agent (2M LA). Note that the LA agent is only relevant when there is nonzero latency, and so $\delta > 0$ for each experiment where the LA agent is present in these experiments.

Parameter	Default	Description
$\bar{r}$	100,000	Mean fundamental value of security.
σ_{shock}^2	5,000,000	Variance of the random shock to the security's fundamental value.
σ_{pv}^2	5,000,000	Variance of ZI private valuations; used to draw private marginal utility of the agent's net position.
α	0.001	Threshold from which the LA agent determines whether an opportunity is worth pursuing.
q_{max}	10	Maximum position (long or short) for ZI traders

Table C.13: Default parameters used across model configurations. These were specified by WW except for α , which is based on the 2013 paper [190] and MarketSim.

Environment	N_{ZIRP}	λ	κ	T	δ
1	24	0.05	0.05	15000	$\{0, 100, 200, 300, 400, 600, 700, 900\}$
2	238	0.005	0.02	10000	$\{0, 50, 100\}$
3	58	0.005	0.02	5000	$\{0, 25, 50, 75, 100\}$

Table C.14: Parameter settings for the three market environments from WW.

As noted previously, the ZIRP traders have the following parameters that define a strategy: $\{R_{\min}, R_{\max}, \eta\}$. Wah and Wellman [189] use an approach called ‘empirical game-theoretic analysis’ (EGTA) to determine the Nash equilibrium mixture of ZIRP strategies for a given experiment, under the assumption that players aim to maximize their expected payoff (surplus) based on the strategies of other players. The allowed

ZIRP strategy parameter settings are shown in Table C.15. The equilibrium strategy mixture specifies the probability of each strategy setting and are given in our supplementary materials. Due to this approach, in the canonical results the trader ecology varies for each combination of environment, market configuration, and latency.

ID	R_{min}	R_{max}	η
ZI_1	0	125	1
ZI_2	0	250	1
ZI_3	0	500	1
ZI_4	250	500	1
ZI_5	0	1000	1
ZI_6	500	1000	0.4
ZI_7	500	1000	1
ZI_8	0	1500	0.6
ZI_9	1000	2000	0.4
ZI_{10}	0	2500	0.4
ZI_{11}	0	2500	1

Table C.15: ZIRP strategy parameter combinations, taken from Wah and Wellman [189] (p. 80).

C.2.3 VERIFICATION OF WW

For our verification exercise, we simulate our implementation of the model for the same experimental design as the canonical study. Wah and Wellman [189] ran each experiment for 500 strategy mixtures, 100 independent runs per mixture—a total of 50,000 runs per experiment. Each mixture is an independent draw of strategies from the strategy profile for that experiment. We repeat this design, drawing 500 mixtures from the strategy profile for the given experiment, as specified in Table C.9. We simulate 100 runs for each mixture. We then plot (Fig. ??) the results for the following metrics, which can then be compared to the canonical figures: mean ZIRP

surplus and LA profits; mean execution time; mean NBBO and BBO spreads; and mean number of transactions. The details of how these metrics are calculated are given in Section C.1.18.

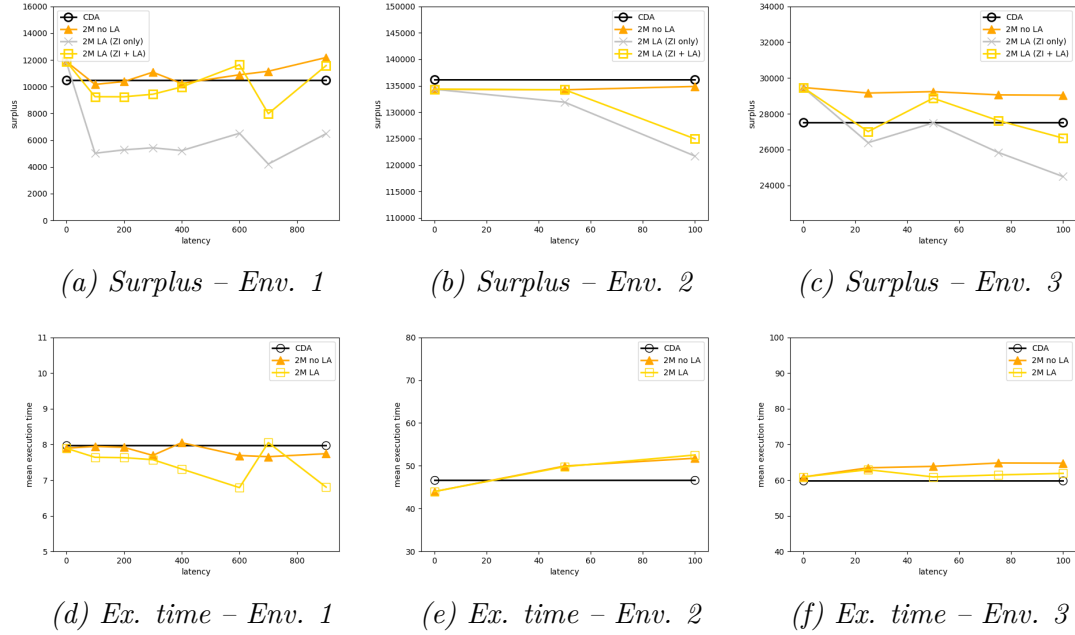
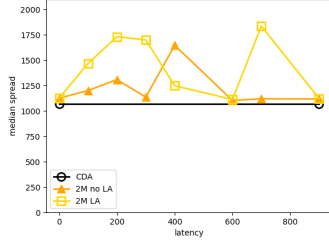
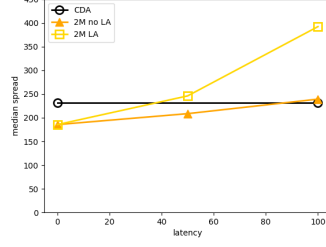


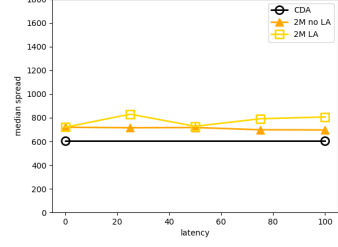
Figure C.1: Surplus and execution time results for our replication of the 2016 Wah and Wellman model [189]. Results are from simulating the model with the settings specified by Wah and Wellman for 500 agent strategy mixtures with 100 runs per mixture (50,000 runs total). Figures should be compared to corresponding results in [189] and Chapter 3 for demonstration of relational equivalence.



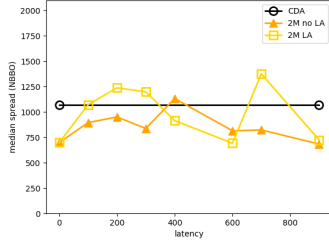
(a) BBO spread – Env. 1



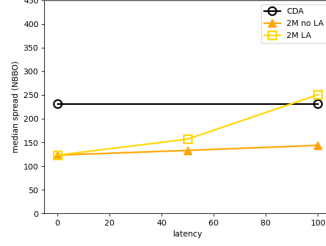
(b) BBO spread – Env. 2



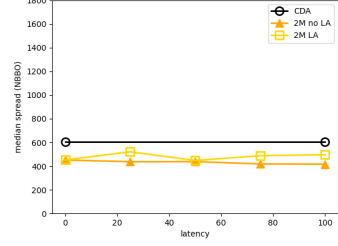
(c) BBO spread – Env. 3



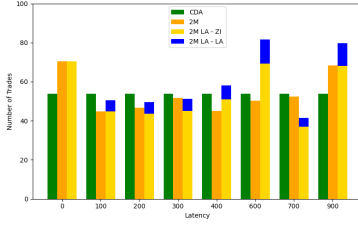
(d) NBBO spread – Env. 1



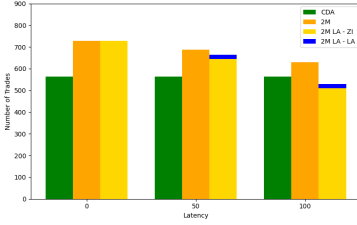
(e) NBBO spread – Env. 2



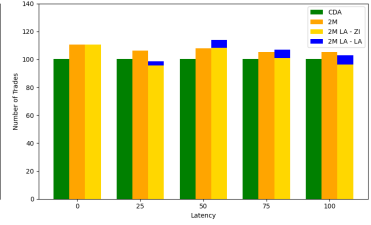
(f) NBBO spread – Env. 3



(g) Transactions – Env. 1



(h) Transactions – Env. 2



(i) Transactions – Env. 3

Figure C.2: Spread and transaction results for our replication of the 2016 Wah and Wellman model [189]. Results are from simulating the model with the settings specified by Wah and Wellman for 500 agent strategy mixtures with 100 runs per mixture (50,000 runs total). Figures should be compared to corresponding results in [189] and Chapter 3 for demonstration of relational equivalence.

C.2.4 VALIDATION OF WW

We next validate the price return behavior generated by the WW model against empirical benchmarks—specifically, we test for lack of linear autocorrelation, heavy tails and aggregational Gaussianity, volatility clustering, volume/volatility clustering, and lack of linear autocorrelation [54]. Our analysis follows similar methodology for each

metric to that detailed in our main paper (Section 4.5). We do not use the experimental design from the original paper (500 mixtures, 100 runs per mixture) for this analysis. Instead, we simulate 1000 mixtures, 1 run per mixture, for each environment. We focus on the single-CDA configuration of the model, using the surplus-maximizing ZIRP strategy profile for this configuration under each environment.

While we use trade prices in our return analysis from the main paper, we calculate the price returns from the Observer DBBO here. The WW experiments on average produce fewer than 1,000 or even 100 trades, depending on the environment [189]. We thus utilize the DBBO mid-price to have a much more less sparse price series to analyze.

Results from 1,000 runs of an experimental setting are pooled together. Analyses that look across time (e.g., correlations and the interarrival times) do not bridge separate samples, however. In other words, the last observation from one sample is not treated as being related in time to the first observation of another sample.

There are two important nuances to discuss prior to detailing the results of statistical analyses on the DBBO. The first relates to how time is formulated in the model. The second feature of the model becomes apparent when we inspect individual DBBO time series. Fig. C.3 shows example DBBO mid-prices and returns for single runs from each of the environments (Envs. 1, 2, and 3), simulated for the equilibrium ZIRP strategy profile corresponding to the respective setting. Note the large price changes at the start of each simulation, with much larger returns in the first time steps than seen throughout the rest of each simulation. The returns are variable but fall within a given range outside of the beginning time steps. We believe the large change at the start may be due to a lack of liquidity in the LOB when the simulation

first begins.

The Env. 1 and 2 simulations also show sharp drop offs in price in the last time steps, likely due to the specifics of the ZIRP strategy and how it values the stock. The ZIRP strategy calculates the expected fundamental value of the stock at the terminal time step T , based on a perfect view of the fundamental value r_t when the trader places its order at time t . As the simulation gets closer to the end of the simulation, the current terminal value is less-likely to revert to the mean before the end of the simulation. As such, orders placed at the end of the simulation will be very sensitive to the fundamental value at the current time step, possibly leading to larger collective price movements (if not particularly large returns at the single-timestep level).

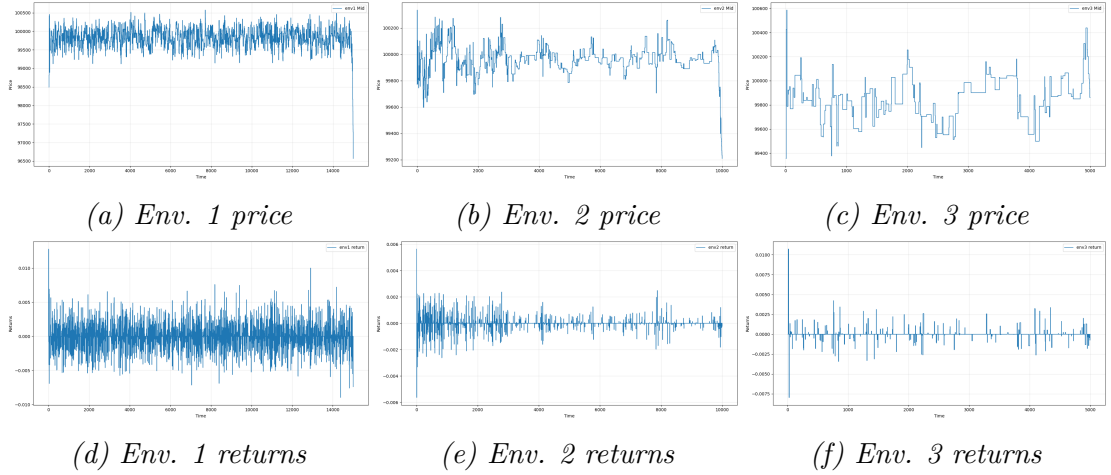


Figure C.3: Example DBBO mid-price and return series from single simulations of each WW environment. Each simulation is from the consolidated-market configuration of the model with the surplus-maximizing equilibrium ZIRP profile for the respective setting.

Regardless of the specific drivers, the overall result is that price returns at the beginning of the simulation differ materially from price returns throughout the rest of the simulation. We test for sensitivity of our analysis to these edge effects by calculating two versions of each metric. The first version considers the full DBBO

series; thus, the large returns at the beginning and end of the simulations are included in the samples. The second version removes the first and last 50 timesteps from each DBBO sample, thus only considering observations for the inner period where behavior appears to be more stable. This will help us determine to what extent these edge effects are driving our results on the examined metrics. One important note for the following metrics is that we do not normalize the returns by time-of-day for the correlation metrics, contrary to our main-paper results (Section 4.5). This is due to the abstract formulation of time (as noted earlier) and the fact that the WW simulations do not have the U-shaped intraday activity pattern present in our extended model.

Absence of autocorrelation

We first calculate the linear autocorrelation of returns. The linear autocorrelation function (ACF) is:

$$C(\tau, \Delta t) = \text{corr}(r(t, \Delta t), r(t + \tau, \Delta t)),$$

taking the Pearson correlation of the return series against a lagged version of itself. In our Chapter 2 results with returns measured in clock-time for $\Delta t = 1Min$, we find ACF for Dow 30 stock returns is less than -0.1 at the first lag and fluctuates around zero beyond that. In Fig. C.4, we show the ACF results for the consolidated-market configuration of Environment 1, using the equilibrium strategy profiles as discussed above. We provide two sets of results: one using the full DBBO time series and the other removing the first and last 50 timesteps. In both sets of results, we find that linear autocorrelation is weak but stays negative for ≈ 50 lags in each environment. Trimming the first and last timesteps appears to increase the magnitude of the ACF

at the first lags while also reducing variation across the strategy profiles.

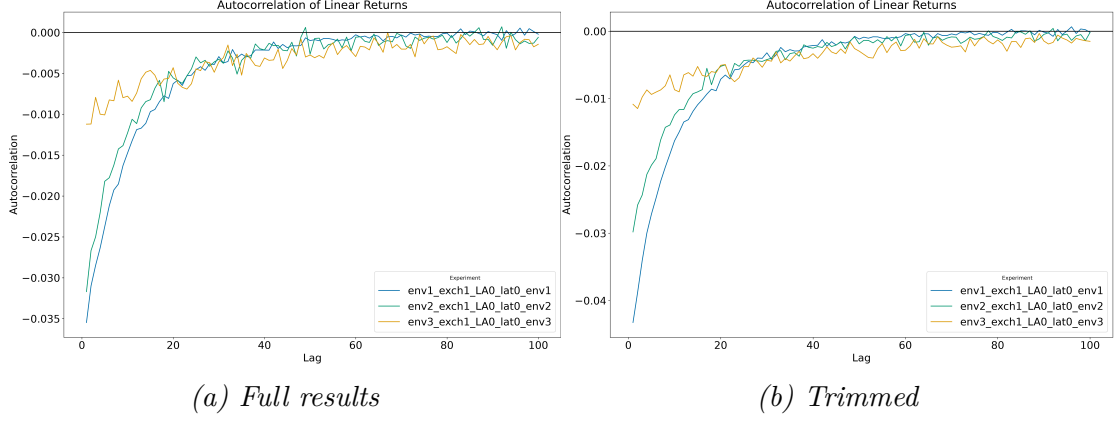


Figure C.4: Linear autocorrelation of returns from the WW model. Results shown are from 1,000 independent runs each for single-market configurations of Environment 1, 2, and 3. Each individual line shows results from the surplus-maximizing equilibrium strategy profile for the single-CDA configuration of the respective environment. The full results do not trim the time series, while the trimmed results remove the first and last 50 time steps of the DBBO before calculating the metric.

Volatility clustering

We next measure volatility clustering by calculating the autocorrelation of the absolute returns:

$$C^2(\tau, \Delta t) = \text{corr}(|r(t, \Delta t)|, |r(t + \tau, \Delta t)|).$$

Empirical results have shown positive squared or absolute autocorrelation and decays slowly, as discussed in Chapter 2. In our empirical results (Figure 2.7), we found positive absolute autocorrelation for more than 100 lags for 1-minute returns and 1,000 lags for trade-level returns. Our results for the WW model (Fig. C.5) show the absolute ACF stays above zero for more than 100 lags. However, the magnitude of the squared ACF is very weak, particularly when we look at the results where the first and last 50 timesteps are removed from each price series. Trimming the timeseries reduces the initial autocorellation level further by about 0.01 for each.

When we compare the absolute ACF results to the linear ACF results, we come to the conclusion that we cannot say the model achieves both a lack of linear autocorrelation and non-trivial volatility clustering. Particularly when we trim the opening and ending timesteps, the absolute ACF looks like the absolute value of the linear ACF. From empirical results, the absolute or squared ACF should be higher at the first lags (≈ 0.1) and decay as approximately a power-law (Section 2.3.4). In log-log scale, the absolute ACF also appears to decay faster than a power-law.

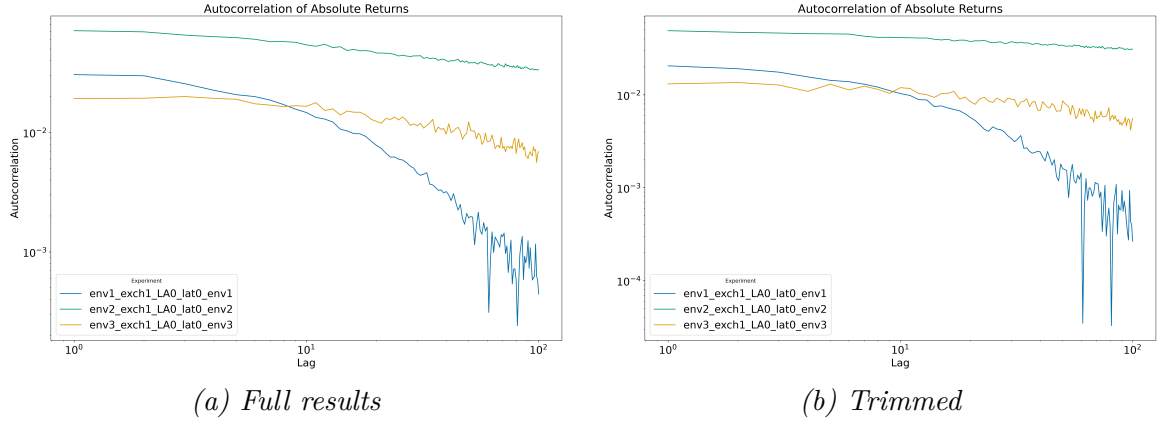


Figure C.5: Autocorrelation of absolute returns for the WW model. Results shown are from 1,000 independent runs each for single-market configurations of Environment 1, 2, and 3. Each individual line shows results from the surplus-maximizing equilibrium strategy profile for the single-CDA configuration of the respective environment. The full results do not trim the time series, while the trimmed results remove the first and last 50 time steps of the DBBO before calculating the metric.

Heavy tails and aggregational Gaussianity

The distribution of asset returns are expected to exhibit heavy tails. As the timescale Δt increases, tail heaviness is expected to decrease, and the distribution approaches a normal distribution. One way to measure this is by looking at the kurtosis of the

return distribution. The excess kurtosis of a distribution is calculated as follows:

$$K(\Delta t) = \frac{\langle (r(t, \Delta t) - \langle r(t, \Delta t) \rangle)^4 \rangle}{\sigma(\Delta t)^4} - 3,$$

where $\sigma(\Delta t)^2$ is the variance of the returns. Note that three is subtracted such that a normal distribution would have $K \approx 0$, while positive kurtosis indicates a distribution is sharper peaked and has heavier tails than a normal distribution. The expectation is therefore that $K(\Delta t)$ should be greater than zero for small timescales and decrease as timescale increases. As noted above, it is not clear what the real-world analog is for a timestep in WW, thus making it difficult to say for what timescales we should expect $K(\Delta t)$ to stay positive.

We plot kurtosis as a function of timescale for the full and trimmed results in Fig. C.6. In the full results, kurtosis is large across the timescales tested for each environment. The Environment 1 results stay flat as timescale increases, while kurtosis decreases as a function of timescale for Environments 2 and 3. When we remove the first and last 50 timesteps, kurtosis is greatly decreased, particularly for Environment 1. We see that $K(\Delta t) < 1$ when $\Delta t \geq 50$ for Environment 1. Note, however, that the simulation length is 15000 timesteps in Environment 1, and the smallest nonzero latency value considered in the Environment 1 experiments is $\delta = 100$. This would mean that, when we remove the opening and closing periods, the return distribution is close to the normal distribution for timescales smaller than the amount of time it takes the SIP to calculate the NBBO in the model. This is inconsistent with the referent system, where the NBBO is calculated on the order of milliseconds or less [64] and kurtosis can stay positive for multiple days [54].

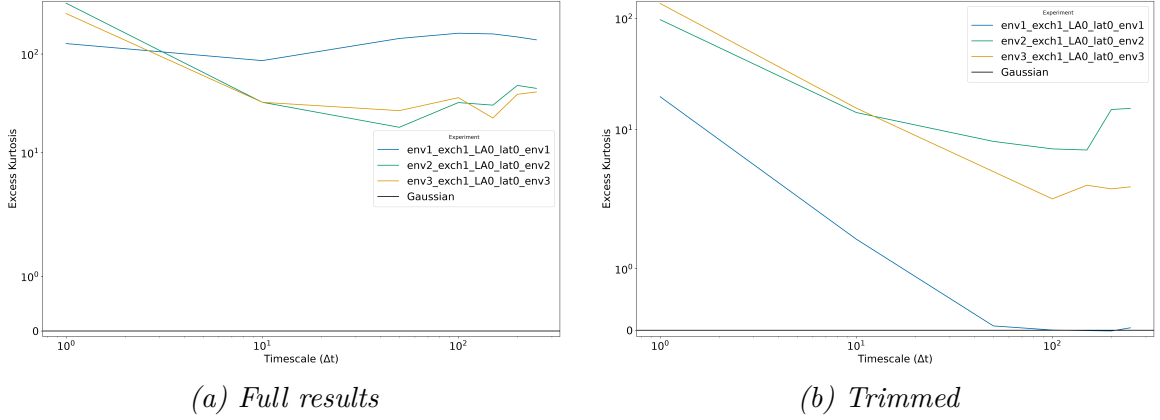


Figure C.6: Kurtosis as a function of timescale for return distributions from the WW model. Results shown are from 1,000 independent runs each for single-market configurations of Environment 1, 2, and 3. Each individual line shows results from the surplus-maximizing equilibrium strategy profile for the single-CDA configuration of the respective environment. The full results do not trim the time series, while the trimmed results remove the first and last 50 time steps of the DBBO before calculating the metric.

Volume/volatility correlation

The final property of returns we examine is the relationship between volume and volatility:

$$VV(\tau, \Delta t) = \text{corr}(|r(t, \Delta t)|, V(t + \tau, \Delta t)).$$

As a reminder, we calculate volume $V(t, \Delta t)$ as the number of DBBO updates received in the period $(t - \Delta t, t]$. The correlation between volume and volatility is expected to be positive, at least synchronously (Section 2.3.7). The results for this metric are shown in Fig. C.7 for the full and trimmed results. We find evidence of the expected relationship in both sets of results, with only slight changes due to removing the first and last 50 timesteps.

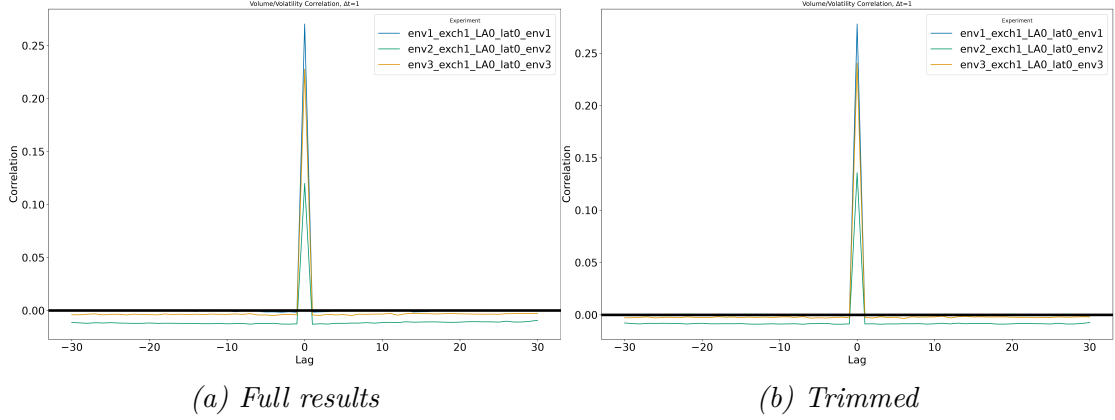


Figure C.7: Volume/volatility correlation for the WW model. Results shown are from 1,000 independent runs each for single-market configurations of Environment 1, 2, and 3. Each individual line shows results from the surplus-maximizing equilibrium strategy profile for the single-CDA configuration of the respective environment. The full results do not trim the time series, while the trimmed results remove the first and last 50 time steps of the DBBO before calculating the metric.

C.3 BASE AND DIVERSE FRAGMENTATION RESULTS

Our main paper (Section 4.5) used *base2x* and *diverse2x* as the main agent configurations for our fragmentation experiment results. In the supplemental results below, we provide similar results for the original *base* and *diverse* agent configurations. While doubling the agents inherently has similar changes to both configurations in terms of number of agents, it arguably makes a larger difference for the *diverse* versus *diverse2x* configurations. Recall that the *diverse* agent configuration has only 1 market maker and 25 momentum agents, introducing asymmetry into the fragmented market settings. This was a main motivation to our focus on the *2x* results in the main-body results.

C.3.1 RAD RESULTS

As in our main results, we begin by plotting the mean RAD results and bootstrap confidence intervals for the consolidated-market baseline versus the other configurations. We find similar high-level takeaways to our main results (Fig. 4.14) in the results for *base* and *diverse* (Fig. C.8). In the NBBO information regime, SIP latency has a larger impact than fragmentation, while in the BBO information regime SIP latency has minimal effect and fragmentation has much larger effects on the price path. Including the LA agent has minimal incremental effects in each set of results.

One difference to highlight is that the RAD values are larger for each of these results than the corresponding results with the *2x* agent configurations. This is particularly true for the *diverse* results with latency under the NBBO information regime relative to the corresponding results from the *diverse2x* agent configuration. The RAD is roughly 0.00175 for *diverse* agent configuration versus roughly 0.0004 in the *diverse2x* results.

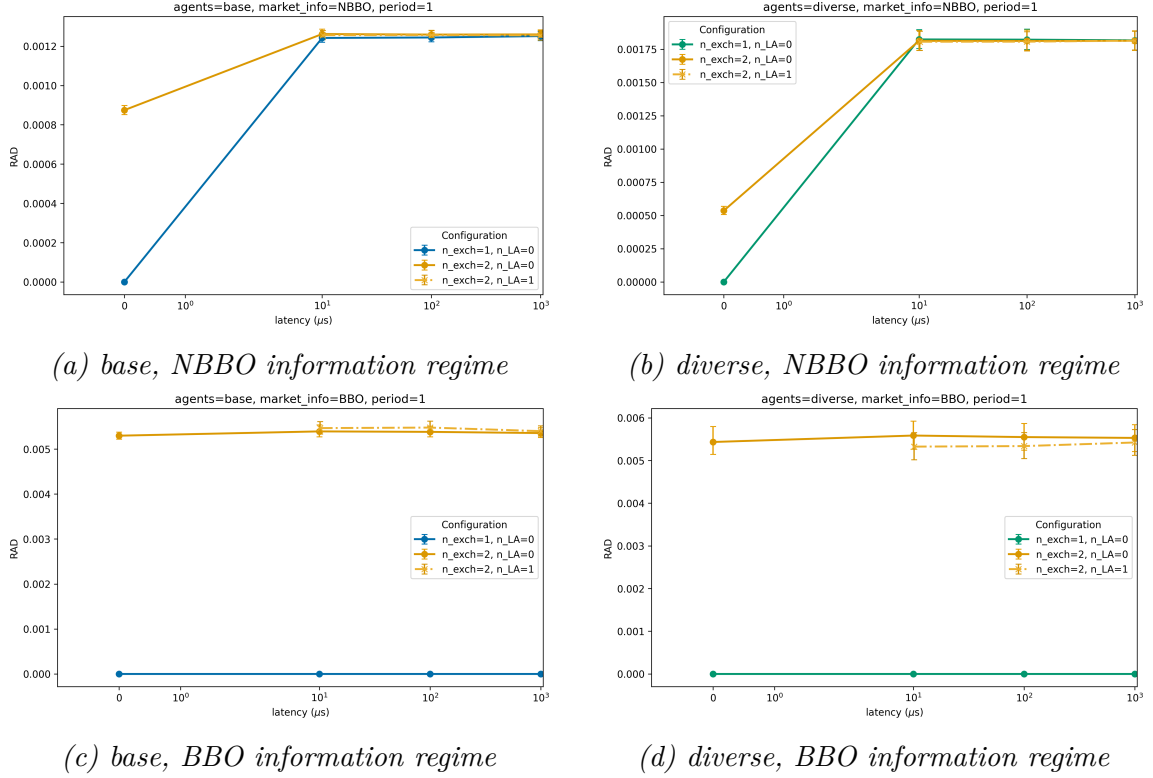


Figure C.8: Relative absolute deviation (RAD) versus consolidated-market baseline without latency.

In the results below, we find similar high-level results to our main takeaways from Section 4.5. Many results show similar qualitative behavior and quantitative ranges between the two sets of results. The main difference worth highlighting is we find differentiation between the results with latency versus without latency in the NBBO information setting for the *diverse* agent configuration, which we did not find for the *diverse2x* configuration. The with-latency results from the *diverse* agent configuration have similar shape and range to the *diverse2x* configurations under the NBBO information regime. However, the without-latency results differ under this information setting, showing lower kurtosis, lower linear and absolute autocorrelation, and negative volume/volatility correlation. Notably, the single-exchange and two-

exchange market configurations perform similarly for similar latency settings under this configuration and information regime.

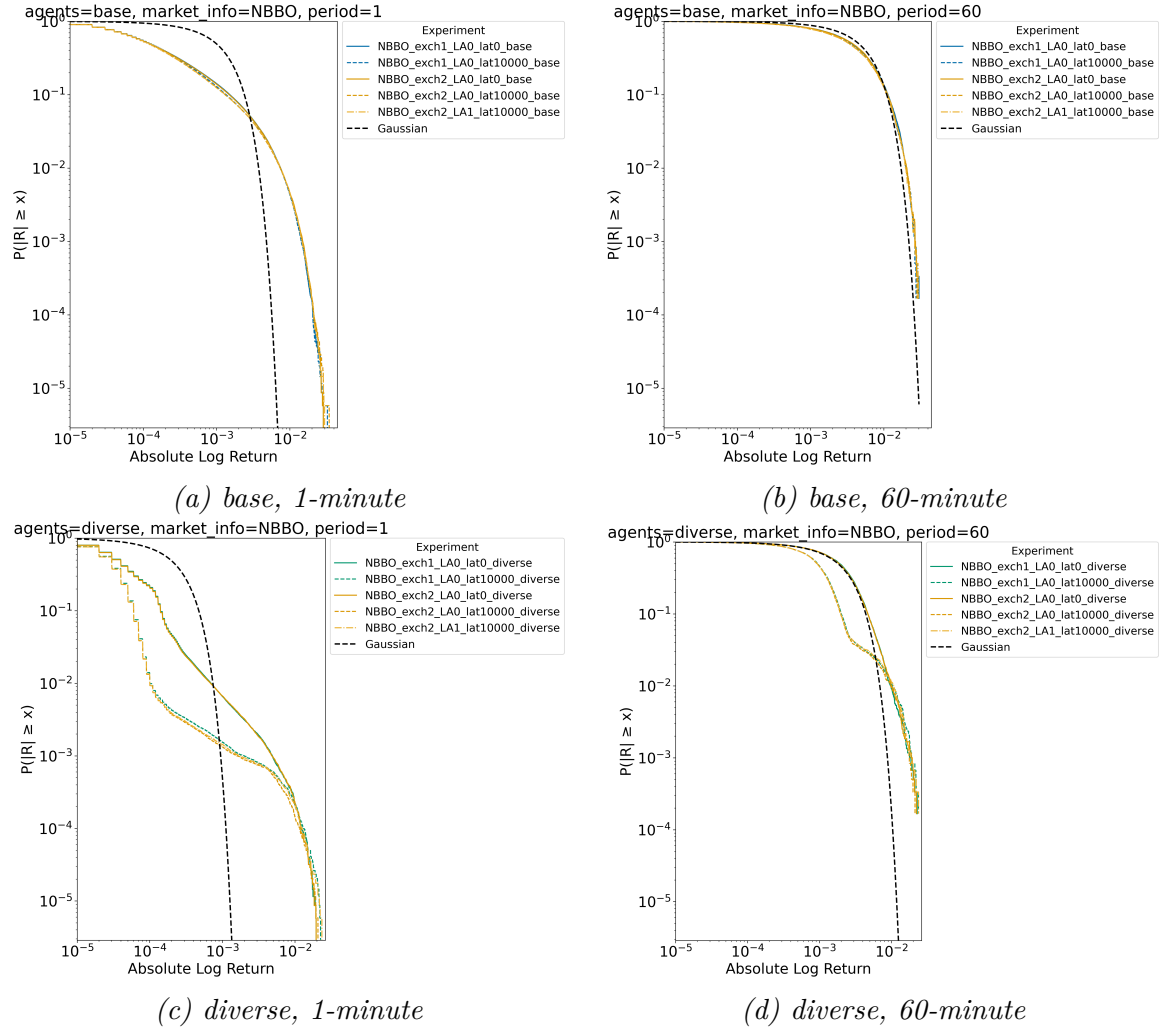


Figure C.9: CCDFs of absolute return distributions for base and diverse agent configurations under the NBBO information regime. Results are from 1000 runs of each experiment. Latency values included in these results are 0 and 10 μ s, written in nanoseconds in the experiment key.

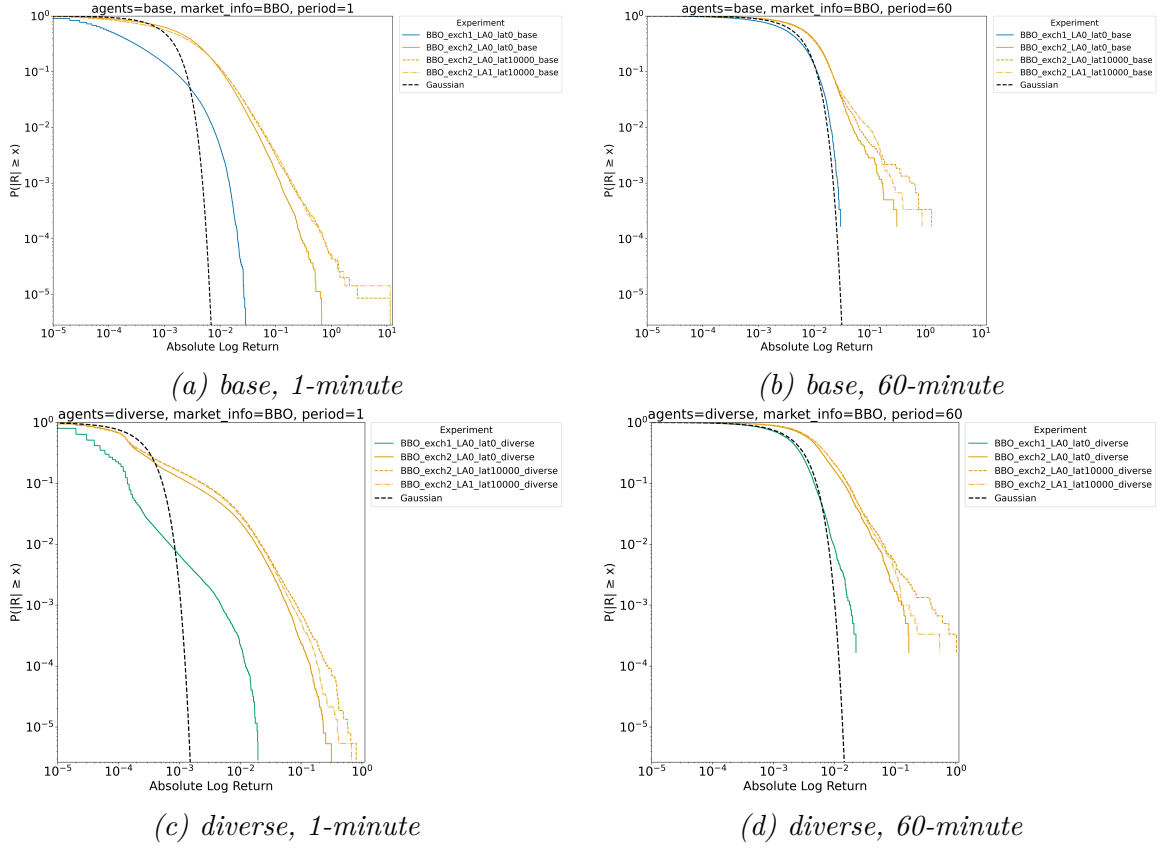


Figure C.10: CCDFs of absolute return distributions for base and diverse agent configurations under the BBO information regime. Results are from 1000 runs of each experiment. Latency values included in these results are 0 and 10 μ s, written in nanoseconds in the experiment key.

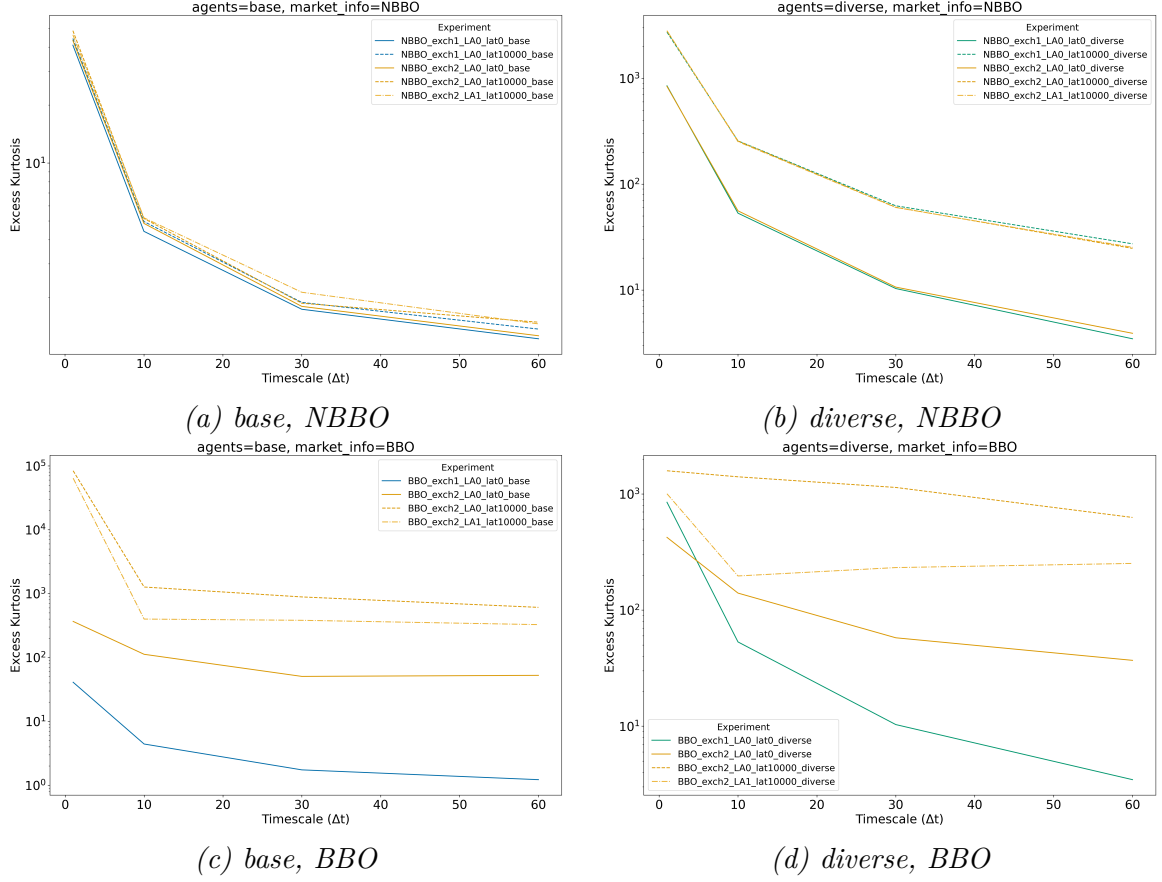


Figure C.11: Kurtosis as a function of timescale for the base and diverse agent configurations. Results are shown for the base and diverse agent configurations under the NBBO and BBO information regimes. Results are from 1000 runs of each experiment. Latency values included in these results are 0 and 10 μ s, written in nanoseconds in the experiment key.

Linear autocorrelation

Next, we measure the linear autocorrelation of returns under the different market settings. As mentioned earlier during the Vyetrenko et al. [187] benchmarking exercise, returns are expected to exhibit minimal linear autocorrelation except at small intraday timescales [54]. The linear autocorrelation function (ACF) is:

$$C(\tau, \Delta t) = \text{corr}(r(t, \Delta t), r(t + \tau, \Delta t)),$$

taking the Pearson correlation of the return series against a lagged version of itself. Cont [54] in 2001 listed $\simeq 20$ minutes as the timescale below which ACF results may be non-trivial due to microstructure effects. More recently, in Chapter 2 we found autocorrelation of 1-minute returns for Dow 30 stocks circa 2018–2019 was less than -0.1 at the first lag and fluctuated around zero for subsequent lags.

In our analysis in Chapter 2, we normalize the empirical return series to account for intraday seasonality before calculating the linear ACF and most other results. Our simulated return series exhibit a similar ‘u-shaped’ intraday pattern of activity, as demonstrated earlier when reproducing the Vyetenko et al. [187] results. We therefore normalize our return series similar to our methodology in Section 2.2, as not doing so may affect the stability of our measure and how directly we can compare to empirical results. We normalize the returns by first centering and standardizing each simulation’s returns by the mean and standard deviation at the given timescale. We then divide each return by the average absolute return at that time of day across the simulations for the given experimental setting.

In the baseline (consolidated-exchange) results and the fragmented results under the NBBO information regime, we find patterns of persistent autocorrelation. This pattern is relatively weak for the *base* agent configuration. Linear ACF from the *diverse* agent configuration stays above 0.1 for more than 40 lags and positive for more than 60 lags, however. This is in contrast to our empirical results of linear ACF in the NMS from Chapter 2, and it is also higher than the corresponding level of absolute autocorrelation found later on in this section. The earlier results from Vyetenko et al. [187] (and our replication, Fig. 4.4) show the distribution of first-lag local autocorrelation was more positively skewed than similar results for historical

stock data. The unrealistic pattern of persistently positive ACF was not apparent from this earlier metric, however. We find that the BBO information regime dampens these effects and brings the level of autocorrelation more in line with empirical results. The first-lag ACF results under the BBO information regime are also negative at the first lag, as is frequently found at intraday timescales [54].

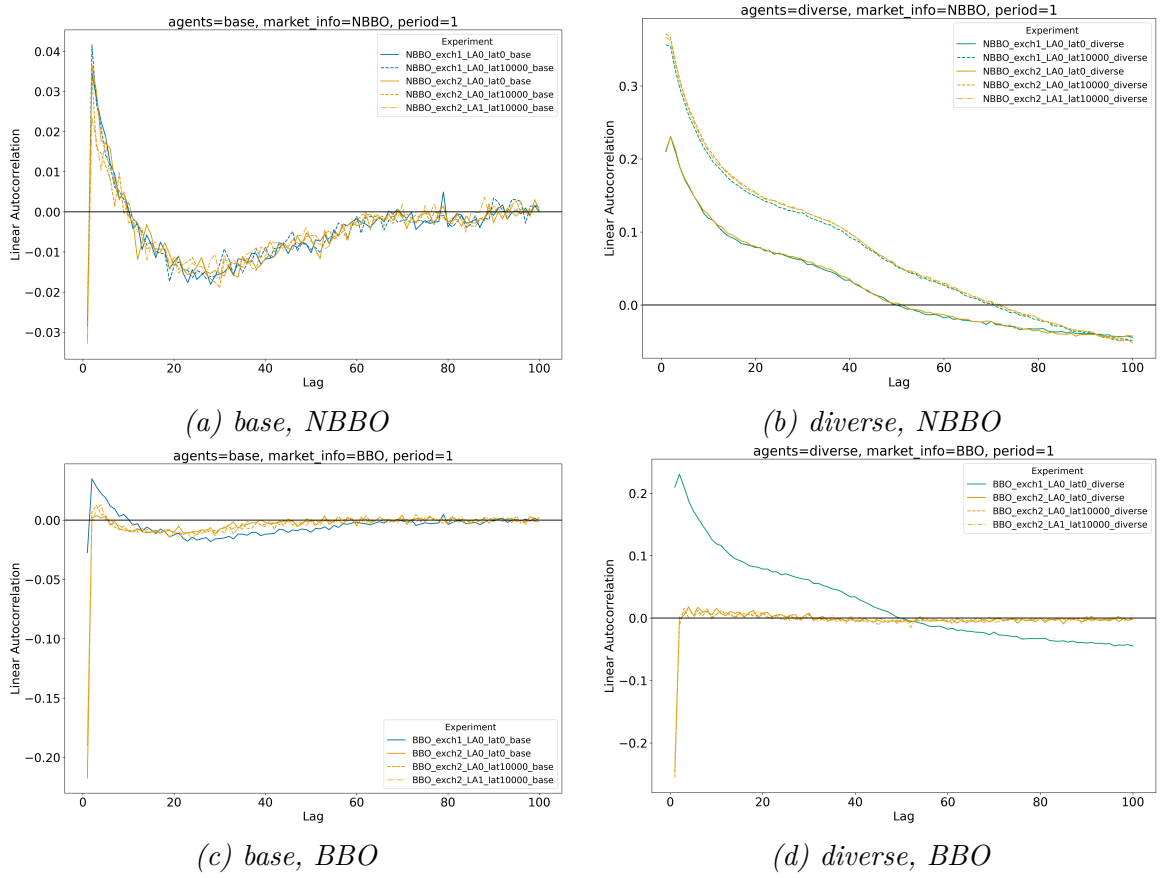


Figure C.12: Linear autocorrelation results for the base and diverse agent configurations with different market information settings. Results are from 1000 runs of each experiment. Latency values included in these results are 0 and 10 μ s, written in nanoseconds in the experiment key.

Volatility clustering

In contrast to the lack of linear autocorrelation, the magnitudes of returns are expected to be positively autocorrelated, with a slow decay as lag increases [54]. We measure these properties here by calculating the autocorrelation of the absolute return series. We again use the normalized return series for this analysis to account for intraday seasonality. We calculate the absolute autocorrelation for up to 100 lags and plot the results in log-log scale here, in order to better assess how the autocorrelation decays as a function of lag.

We overall find evidence of volatility clustering and slow-decay of absolute autocorrelation for each set of results (Fig. C.13). The level of absolute autocorrelation found and the shape of its decay as a function of lag are influenced by the structural settings, particularly when comparing the consolidated- and fragmented-market results under the BBO information regime. Fragmentation under this information setting greatly increases the absolute autocorrelation and produces a slower decrease as a function of lag. The fragmented markets without latency for both agent configurations produces absolute autocorrelation values that start and remain much higher than empirical levels found for similar metrics (see Section 2.3.4). Latency further increases the absolute autocorrelation for the *base* agent configuration but dampens it for the *diverse* agent configuration. Each curve exhibits a sharp drop-off eventually, however, rather than the expected power-law decay [54]. A true power-law decay is not possible when measuring intraday returns at a finite timescale—there are only 390 minutes in the trading day—but most stocks in the sample we analyzed in Chapter 2 had positive 1-minute absolute autocorrelation for ≈ 100 lags or more.

Under the NBBO information regime, the market configurations with latency

and/or fragmentation stay close to the consolidated-market baseline, at least for the first ≈ 20 lags. The *diverse* curves show an interesting shape beginning around this point, and the fragmented-market absolute autocorrelation actually decreases faster than under the consolidated-market settings for this agent configuration. Much less differentiation is found across the *base* results for the NBBO information regime.

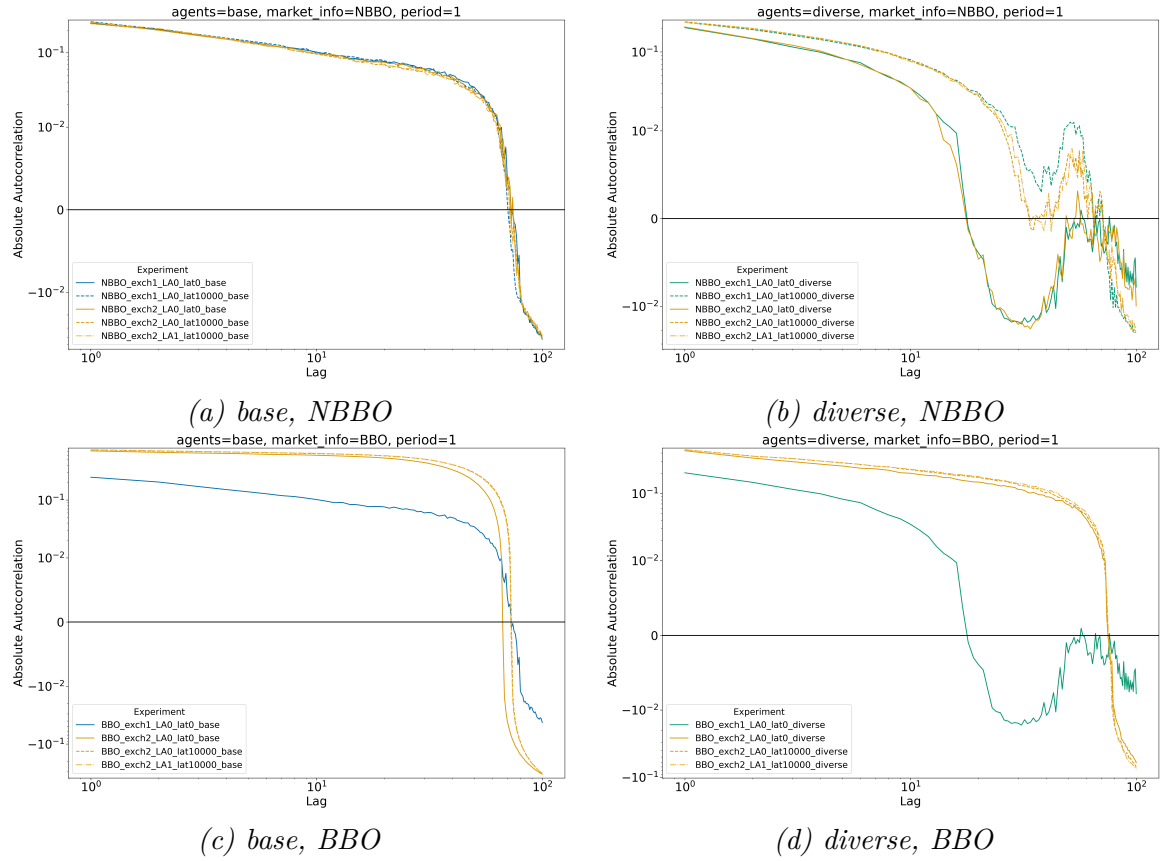
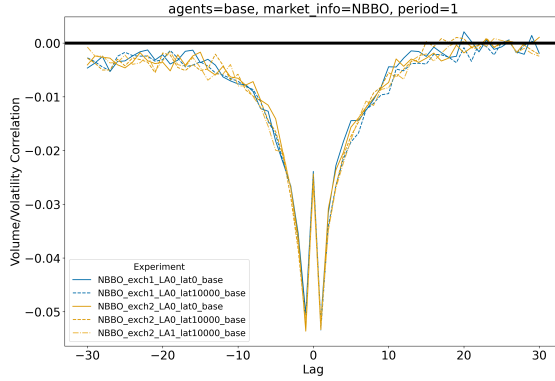


Figure C.13: Absolute autocorrelation results for the base and diverse agent configurations with different market information settings. Results are from 1000 runs of each experiment. Latency values included in these results are 0 and 10 μ s, written in nanoseconds in the experiment key.

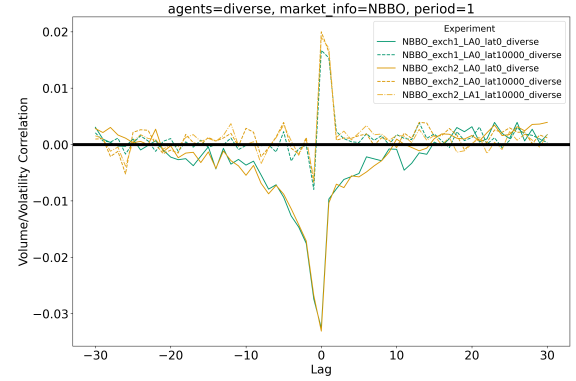
Volume-volatility correlation

Finally, we calculate the correlation between volume and volatility, which are expected to be positively related, at least synchronously (see Section 2.3.7). We measure this property here by calculating the correlation for all samples under a given experimental setting, versus the sample-level correlation calculated for our alignment exercise against Vyetenko et al. [187] (Section 4.4.1). We normalize the volume series by the intraday activity pattern, analogous to how we normalize the returns and volatility series. We center and standardize the volume from each simulation before dividing the volume at a given time by the average amount of volume at that time across the simulations for that experiment.

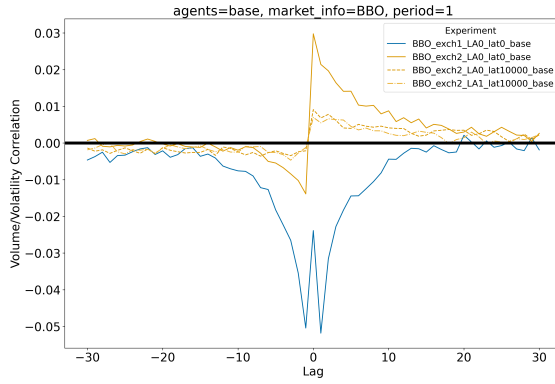
Overall, the volume-volatility correlation is found to be relatively weak compared to empirical results. In many of the experimental results, the synchronous (zero-lag) correlation is negative, in contrast with the expected relationship. Fragmentation alters the strength of the correlation under the BBO information regime, but this effect does not necessarily bring the correlation more in line with empirical results. In the *diverse* agent configuration results, the synchronous correlation between volume and volatility is positive for each setting but weaker in the fragmented-market results than for the consolidated-market setting. Fragmentation meanwhile changes the relationship from negative to weakly positive for the *base* agent configuration under the BBO information setting. Under the NBBO information setting, fragmentation and latency have little effect for the *base* agent configuration, whereas nonzero latency increases the volume/volatility correlation for the *diverse* agent configuration.



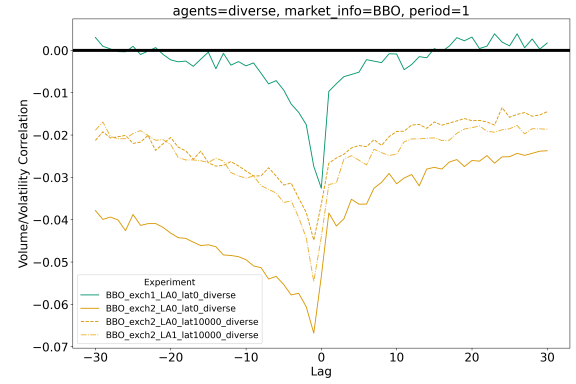
(a) base, NBBO



(b) diverse, NBBO



(c) base, BBO



(d) diverse, BBO

Figure C.14: Volume-volatility correlation results for the base and diverse agent configurations with different market information settings. Results are from 1000 runs of each experiment. Latency values included in these results are 0 and 10 μ s, written in nanoseconds in the experiment key.