CHAD: A Scalable Turn-Based Simulator for Complex Heterogeneous Agent Dynamics

Abstract

We present CHAD (Complex Heterogeneous Agent Dynamics), an agent-based market simulator implemented in Julia that addresses scalability and reproducibility limitations of existing frameworks. CHAD introduces two principal design contributions: (i) a pre-scheduled, turn-based architecture that precomputes all agent activation times, enabling deterministic execution and near-linear computational scaling to millions of agents; and (ii) a dual-book system comprising a public Limit Order Book and a private Stop-Loss Order Book that endogenously produces cascading liquidity dynamics. Through a combinatorial grid search over 784 agent population configurations, we demonstrate that—within the tested parameter regime—macroscopic market properties including fat-tailed return distributions, volatility clustering, and rapid cascading liquidity events are primarily driven by agent population composition. Validation against empirical BTC/USDT data confirms that the simulator reproduces a broad range of stylized facts. Computationally, the combination of a pre-scheduled architecture and Julia's JIT compilation enables stable execution at one million agents, a scale at which the event-driven ABIDES framework could not complete. We conclude with open questions on calibration methodology and the generalisability of these ecological dynamics beyond financial markets.

Keywords: Agent-based simulation, multi-agent systems, market microstructure, emergent dynamics, scalability, heterogeneous agents

1 Introduction

Financial markets offer a demanding testbed for multi-agent simulation: the statistical regularities of market returns—known as *stylized facts*—are well-documented and provide quantitative benchmarks against which simulated dynamics can be assessed [5]. Pioneering work by Lux and Marchesi [14] showed that these regularities, including fat-tailed return distributions, volatility clustering, and the absence of linear autocorrelation, can emerge from heterogeneous agent interactions alone.

Several established agent-based platforms have advanced the study of market dynamics [17], most notably ABIDES [3], the Bristol Stock Exchange [4], and MAXE [1]. These systems typically employ event-driven architectures with dynamic priority queues and asynchronous scheduling. While flexible, such designs can limit scalability and complicate reproducibility at large agent counts. Moreover, most existing simulators omit stop-loss orders—a class of hidden, conditional orders that empirical evidence links to liquidity cascades in currency markets [16]. It was shown [13] that LOB structure and stop-loss cascades can largely determine price dynamics and produce heavy-tailed return distributions; however, the simulator in that study depended on historical order book data and had limited computational scalability.

This paper introduces CHAD (Complex Heterogeneous Agent Dynamics), a simulator designed to overcome these limitations. CHAD's contributions are:

1. A pre-scheduled, turn-based architecture that precomputes all agent activation times, eliminating queue-management overhead and guaranteeing deterministic execution.

2. A dual-book system comprising a public LOB and a private Stop-Loss Order Book (SLOB), enabling realistic modelling of hidden liquidity and recursive stop-loss cascades.

3. A heterogeneous agent ecology with configurable archetypes, demonstrating that—within the tested parameter regime—macroscopic market properties are primarily driven by agent population composition.

4. Validation against empirical BTC/USDT stylized facts using a validity corridor methodology, and comparative benchmarking against ABIDES.

5. A sensitivity analysis revealing the structural determinants of market realism, including phase transitions and a "fragility pocket" where agent timescales synchronise with cascade dynamics to produce flash-crash-like events.

The architectural pattern, the agent ecology framework, and the validation methodology are not specific to financial markets and may apply to other MABS domains.

2 Architecture

2.1 Pre-Scheduled, Turn-Based Design

Unlike traditional discrete-event simulators that manage a dynamic priority queue, CHAD computes a master schedule of all agent decision points at initialisation. A scheduling function τ computes, for each agent, the set of timestamps $T = \{t_1, t_2, \ldots, t_n\}$ at which it will be activated. The simulation engine then advances sequentially through these time steps, activating only the relevant agents at each step.

Pre-computing the schedule has two practical consequences. First, execution order is fully deterministic: when multiple agents share an activation time t_i, a secondary sort by agent ID resolves ties at initialisation rather than at runtime. Second, the schedule is a flat sorted array rather than a dynamic priority queue, so advancing the simulation reduces to iterating through the array without insertion or deletion overhead. Agents can be assigned diverse wake-up intervals to model a range of operational tempos, from high-frequency to slow retail participants.

This architecture intentionally trades microsecond-level asynchronous fidelity for macro-level performance. It is not designed for research on latency arbitrage or network delays, but for studying emergent phenomena from large heterogeneous populations where reproducibility and high-throughput iteration matter more than event-level timing.

2.2 Dual-Book System: LOB and SLOB

CHAD models a continuous double-auction market [10] around two data structures: a public Limit Order Book (LOB) and a private Stop-Loss Order Book (SLOB). The LOB follows the standard stochastic framework of Cont et al. [6], while the SLOB is motivated by the empirical findings of Osler [16] on stop-loss clustering in currency markets.

Let $\mathcal{L}_t$ and $\mathcal{S}_t$ denote the states of the LOB and SLOB at time t, and let P_t^{last} denote the last traded price. The best bid and ask are:

$$B_t = \max\{p \mid (\text{buy}, q, p) \in \mathcal{L}_t\}, \quad A_t = \min\{p \mid (\text{sell}, q, p) \in \mathcal{L}_t\} \tag{1}$$

The simulator supports market orders (immediate execution at the best available price), limit orders (execution at a specified price or better, with any remainder resting on the LOB), and stop-loss orders (held privately in the SLOB and triggered when the market price crosses a specified level).

The critical mechanism is the *recursive stop-loss cascade*. When an incoming order produces a trade and updates the market price, the system checks the SLOB for any triggered stop-loss orders. Triggered orders are converted into market orders and executed against the LOB, potentially producing further price changes that trigger additional stop-loss orders. This continues recursively until no further triggers remain:

$$\forall\, O \in \mathcal{S}_{t+k}, \quad \text{Trigger}(O) = \text{False} \tag{2}$$

Figure 1 illustrates the complete order execution logic, including the recursive cascade loop.

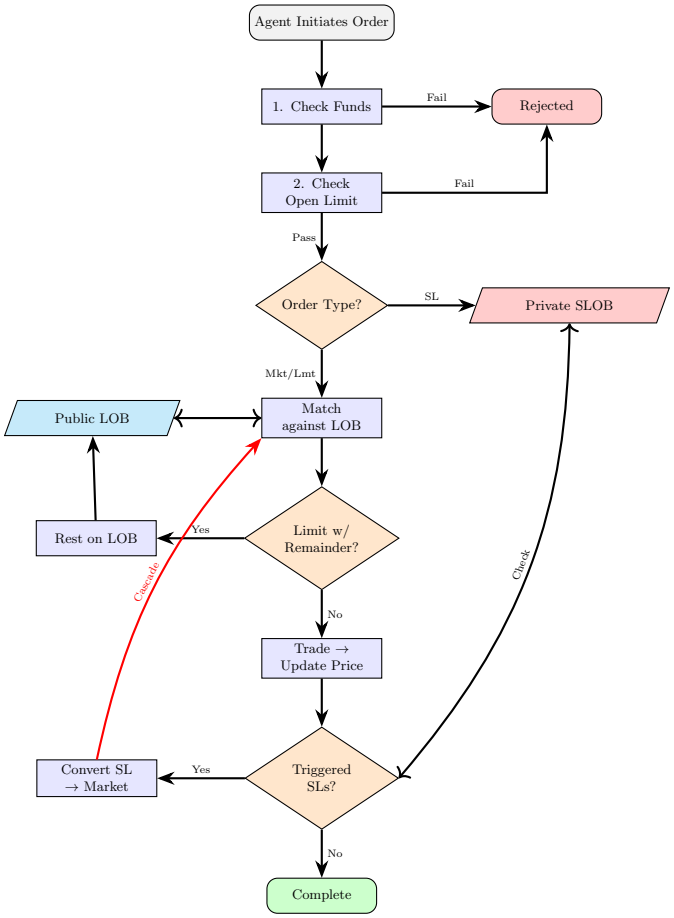


Figure 1: Order execution logic with recursive stop-loss cascade. Triggered stop-loss orders re-enter the matching engine as market orders, potentially triggering further cascades.

3 Agent Design and Heterogeneity

Each agent is an autonomous entity with a unique identity, cash and share holdings, and a flexible parameter dictionary that defines its trading behaviour. Agents are differentiated along several dimensions: trading strategy, decision frequency, trade size, look-back period, signal sensitivity, maximum concurrent orders, and propensity to use stop-loss orders. Table 1 summarises the parameter distributions used to generate heterogeneous populations.

This parameter diversity prevents artificial synchronisation. Heterogeneous decision frequencies ensure that agents do not all act simultaneously; varied look-back windows create a mix of short-term and medium-term time horizons; and distributed trade sizes ensure that price changes emerge from collective order flow rather than individual large orders.

3.1 Agent Archetypes

CHAD supports a modular set of agent archetypes. Let p_{t-1}^{market} denote the last observed market price, p_t^{oracle} a fundamental price from an external series, and $\text{SMA}_N(t) = \frac{1}{N}\sum_{i=1}^{N} p_{t-i}^{\text{market}}$ the simple moving average over look-back period N. The archetypes are:

Value Agents trade toward a fundamental price: buy when $p_{t-1}^{\text{market}} < p_t^{\text{oracle}}$, sell otherwise.

Stochastic Agents provide random liquidity in the tradition of zero-intelligence traders [9, 7]: direction is chosen with equal probability, and order price is a small random perturbation of the current market price.

Table 1: Agent parameter distributions for the sensitivity analysis.

Parameter	Description	Distribution
`sample_freq`	Decision frequency	$\mathcal{U}(25, 250)$
`trade_size`	Order size (fraction)	Disc. $\mathcal{U}\{0.01, \ldots, 0.10\}$
`N`	Look-back period	$\mathcal{U}(25, 500)$
`sensitivity`	Signal threshold	$\mathcal{U}(1, 40)$
`max_open`	Max concurrent orders	$\mathcal{U}(1, 2)$
`stop_loss_prob`	SL placement probability	$\mathcal{U}(0.0, 0.3)$

Table 2: Empirical validity corridor (BTC/USDT daily bounds, 2017–2024).

Metric	Global Mean	Daily Mean	5% Bound	95% Bound
Kurtosis	170.5922	23.0449	2.8434	81.0102
Skewness	1.8071	0.8906	−1.1181	3.7826
Volatility	0.0011	0.0009	0.0003	0.0021
Hurst Exp.	0.5335	0.5832	0.5299	0.6294
GARCH Persist.	1.1299	1.0762	0.8841	1.7303
Vol Clustering	0.3768	0.2116	0.0847	0.3903
Return ACF	0.0331	0.0976	−0.0897	0.2132

Heuristic Stochastic Agents extend the stochastic model with a simple rule to avoid immediate losses, constraining the order price to be advantageous relative to the current market price.

Trend-Following Agents use a momentum strategy, buying when $p_{t-1}^{\text{market}} > \text{SMA}_N(t)$ and selling otherwise, submitting market orders.

Mean-Reverting Agents employ the inverse logic, buying when the price is below the moving average and selling when above.

Opportunistic Liquidity Providers are activated after large price shocks and probabilistically provide counter-directional liquidity with probability $\beta \in (0.5, 1]$.

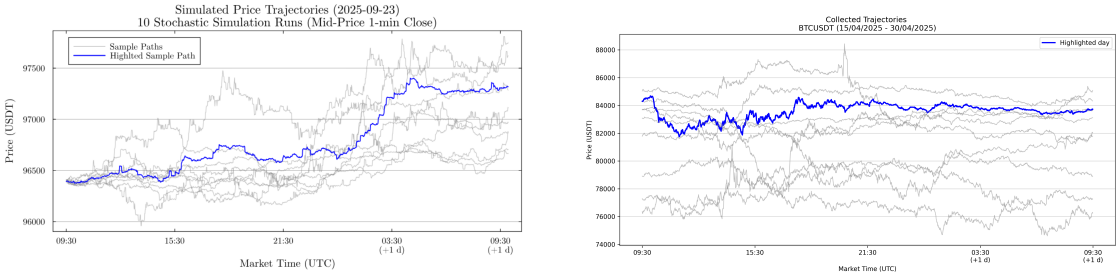
Trade sizes can be fixed, proportional to portfolio value, or drawn from a probability distribution. In our experiments, we found that a Burr distribution with coefficients c=0.776 and d=2.755 closely replicates empirical BTC/USDT trade size distributions.

4 Validation

4.1 The Validity Corridor

A common challenge in validating agent-based simulators against historical data is non-stationarity: comparing a short-run simulation to a multi-year average can be misleading if the global average is distorted by regime shifts [5]. To address this, we decompose high-frequency BTC/USDT data (1-minute candles, 2017–2024) into rolling 24-hour windows and define a *validity corridor*: a simulation is considered empirically plausible with respect to a stylized fact F if F falls within the 5th and 95th percentiles of the daily distribution. We note that this corridor is deliberately broad (e.g., kurtosis bounds of $[2.84, 81.0]$) and functions as a *necessary* condition for realism rather than a sufficient one; the moment-matching loss function $J(\theta)$ described in Section 5 provides the finer-grained discriminator.

Table 2 presents the empirical bounds. Notably, the global kurtosis (170.6) is an outlier relative to the daily regime (23.0), illustrating why the daily-window approach is preferable.



(a) Simulated

(b) Historical BTC/USDT

Figure 2: Comparison of a simulated price trajectory (a) against a historical BTC/USDT sample (b) at 1-minute resolution. Both series display similar intermittency and volatility structure.

Table 3: Baseline stylized facts (100 runs, 8,101 agents).

Fact	**Statistic**	**Mean**	**Std**	**Literature**
Kurtosis	Excess Kurt.	5.9012	0.1076	4–10+
Skewness	Skewness	−0.0126	0.0202	−0.5 to 0.1
Leverage	Corr. coeff.	−0.0002	0.0037	−0.1 to −0.6

4.2 Baseline Validation: Time-Series Stylized Facts

We conducted 100 independent Monte Carlo runs using a baseline population of 8,101 agents (2,000 Stochastic, 5,000 Heuristic Stochastic, 100 Market Makers, 500 Trend-Followers, 500 Mean-Reverters, and 1 oracle agent). Log-returns $r_t = \ln(P_t/P_{t-1})$ were computed from 1-minute aggregated price candles.

Table 3 summarises the baseline results. The simulated returns exhibit a mean excess kurtosis of approximately 5.90, confirming that the simulator generates heavy tails consistent with [15]. A slight negative skewness (−0.013) is consistent with the observation that market crashes are statistically more probable than equivalent upward surges. The leverage effect is effectively absent in this baseline configuration (−0.0002), indicating that the volatility-feedback loop requires tuning of agent population ratios, which we address in Section 5. Figure 2 provides a qualitative comparison: the simulated and historical price series exhibit similar intermittency and volatility structure at 1-minute resolution.

We also compared the simulated LOB against empirical BTC/USDT Level-2 data from Bybit, using 96 order book snapshots (32 per day across three non-consecutive trading days, spaced to avoid intraday autocorrelation). The simulated bid-ask spread (0.111) is close to the empirical mean (0.102), but the simulated book shows stronger sell-side imbalance (−0.243 vs −0.048), suggesting the baseline ecology overproduces selling pressure.

5 Sensitivity Analysis and Emergent Dynamics

5.1 Combinatorial Grid Search

To assess the model's tunability, we deployed a combinatorial grid search across 784 "agent ecologies," defined as vectors $E = [N_{\text{Stoch}}, N_{\text{Trend}}, N_{\text{MeanRev}}, N_{\text{MM}}]$ with $N_{\text{Stoch}} \in \{1, 500, 1000, 2000, 3000, 3500, 4000, 5000\}$, $N_{\text{Trend}} \in \{0, 10, 50, 100, 200, 500, 1500\}$, $N_{\text{MeanRev}} \in \{0, 10, 50, 100, 200, 500, 1500\}$, and $N_{\text{MM}} \in \{0, 1\}$. In all configurations, the Heuristic Stochastic agents (5,000) and oracle agent (1) were held fixed at their baseline values; only the four population counts above were varied.

Each ecology was run 5 times with unique random seeds. Preliminary convergence checks confirmed that inter-configuration variance was an order of magnitude larger than intra-configuration variance, providing sufficient statistical power.

We evaluate each configuration using a moment-matching loss function inspired by the Method

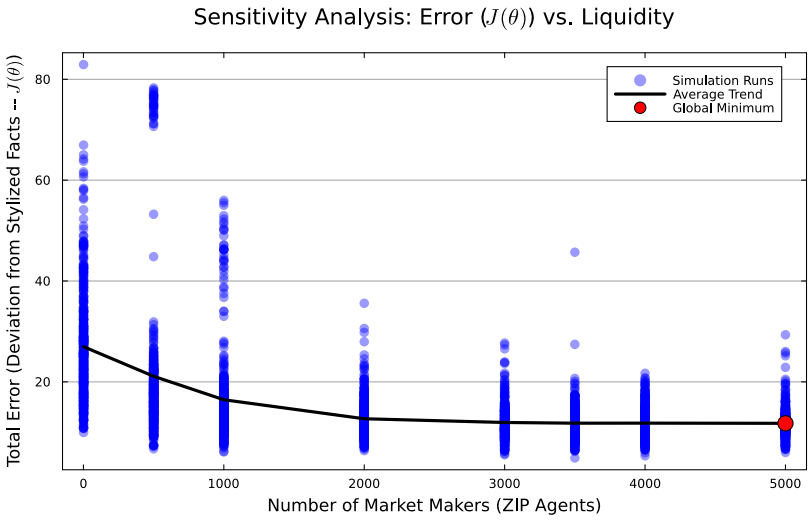


Figure 3: Aggregate error $J(\theta)$ vs. stochastic agent population. The mean trend (black line) decays steeply below $N_{\text{Stoch}} \approx 1{,}000$ and plateaus beyond $\approx 2{,}000$. The red dot marks the global minimum. Each point represents one of 784 population configurations (5 runs each).

of Simulated Moments [8]:

$$J(\theta) = \sum_{i=1}^{K} w_i \left| \frac{m_{\text{sim},i}(\theta) - \mu_{\text{data},i}}{\sigma_{\text{data},i}} \right| \tag{3}$$

where $m_{\text{sim},i}$ is the simulated moment, $\mu_{\text{data},i}$ and $\sigma_{\text{data},i}$ are the empirical mean and standard deviation, and w_i is a weight (doubled for kurtosis and the Hurst exponent to prioritise non-linear signatures). The function aggregates 11 moments: kurtosis, skewness, volatility, Hurst exponent, GARCH persistence, volatility ACF at lags $\{1, 5, 10, 20\}$, and return ACF at lags $\{1, 5\}$.

5.2 Three Liquidity Regimes

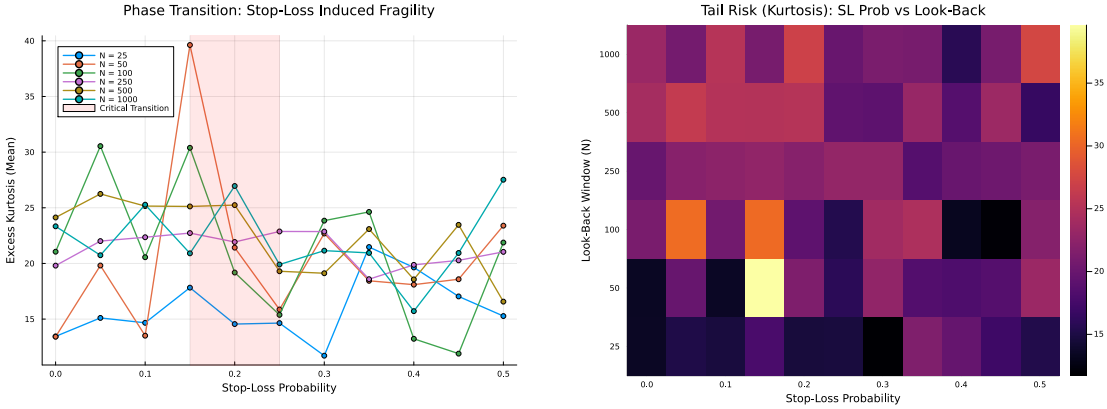
The dominant factor in model fidelity is the population of stochastic liquidity providers (N_{Stoch}, labelled "Number of Market Makers" in Figure 3). Figure 3 shows the relationship between N_{Stoch} and the aggregate error $J(\theta)$, which reveals three regimes. Below $N_{\text{Stoch}} \approx 1{,}000$, a *liquidity crisis* produces high mean error (above 20) with large variance (individual runs exceeding 80): insufficient market depth fails to absorb order flow, resulting in brittle price formation. In the *plausible zone* ($1{,}000 \leq N_{\text{Stoch}} \leq 3{,}500$), error decays rapidly as the added depth smooths price discovery and allows the system to reproduce volatility clustering and fat tails. Beyond $N_{\text{Stoch}} \approx 4{,}000$, the mean error plateaus but does not improve further, suggesting that additional liquidity no longer increases realism.

A polynomial regression of normalised agent populations against $J(\theta)$ (Table 4) confirms these dynamics quantitatively. The stochastic agent population acts as the primary stabiliser ($\beta = -4.50$, $p < 10^{-99}$), with a significant positive quadratic term ($\beta = 3.14$) confirming diminishing marginal returns. Trend followers are a significant linear destabiliser ($\beta = 3.21$). Notably, within this experimental setup, mean-reverting agents are statistically insignificant ($p = 0.44$). One interpretation is that counter-trend pressure is adequately supplied by the LOB structure itself, via the limit orders of stochastic agents—consistent with the finding of Farmer et al. [7] that zero-intelligence order flow can reproduce key market regularities; however, this finding may also reflect the dominance of 3,000 stochastic agents in the tested ecologies, and we caution against generalising it beyond this specific configuration.

We note an important caveat: the grid search varies only population composition while holding internal agent parameters (Table 1) fixed. The sensitivity of results to internal parameter distributions (e.g., look-back windows, trade sizes) remains a direction for future work, though

Table 4: Polynomial sensitivity regression: drivers of simulation error.

Factor	**Coef**	**t-stat**	p	**Role**
Intercept	12.38	67.18	$< 10^{-99}$	Baseline
z_{Stoch}	−4.50	−38.61	$< 10^{-99}$	Stabiliser
z^2_{Stoch}	3.14	21.96	$< 10^{-99}$	Diminishing returns
z_{Trend}	3.21	27.58	$< 10^{-99}$	Destabiliser
z_{MeanRev}	0.09	0.77	0.44	Insignificant



(a) Phase transition by look-back N

(b) Fragility pocket heatmap

Figure 4: Kurtosis response to stop-loss density and agent memory. (a) The shortest look-back windows (N=25, N=50) produce kurtosis peaks at moderate stop-loss densities; longer memories diminish the effect. The shaded region marks the critical transition zone. (b) Heatmap showing the fragility pocket (bright cells) where agent reaction times synchronise with cascade frequencies.

the stop-loss analysis in Section 5.3 provides partial evidence that internal parameters can significantly affect dynamics in specific regimes.

The configuration selected from the grid search ($N_{\text{Stoch}} = 3{,}000$, $N_{\text{Trend}} = 50$, $N_{\text{MeanRev}} = 10$, $N_{\text{MM}} = 1$) achieves a simulated kurtosis of 22.97 against the empirical daily mean of 23.04. We stress that this is a *selected* configuration from 784 tested ecologies, not an independent out-of-sample validation; the close match on kurtosis reflects the search process. Across the plausible zone ($1{,}000 \leq N_{\text{Stoch}} \leq 3{,}500$), kurtosis values ranged from approximately 5 to 30, indicating that the model can be tuned to target specific market regimes rather than being locked to a single point.

5.3 Emergent Fragility: Stop-Loss Cascades

A key finding concerns the interaction between agent memory (look-back window N) and stop-loss density (ρ_{SL}). We performed a controlled sweep across $\rho_{\text{SL}} \in [0.0, 0.5]$ and $N \in \{25, 50, 100, 250, 500, 1000\}$.

For agents with the shortest look-back windows ($N = 25$), the simulator reproduces a distinct "resonance effect" peaking at $\rho_{\text{SL}} \approx 0.20$, where kurtosis reaches approximately 40 (Figure 4a). Agents with $N = 50$ show a similar but weaker peak (kurtosis ≈ 30) at lower stop-loss density ($\rho_{\text{SL}} \approx 0.10$). In both cases, a small cascade causes short-memory agents to rapidly update their trend signals, amplifying the shock and generating flash-crash-like dynamics from simple micro-rules. The heatmap in Figure 4b reveals the spatial extent of this fragility pocket: it is concentrated at short agent memories and moderate stop-loss densities. We note that while the kurtosis spikes and the rapid price trajectories (Figure 5) are consistent with flash crash phenomenology [11], we have not formally characterised these events in terms of duration and recovery metrics. The effect diminishes with longer look-back windows and is largely absent at $N = 1000$, where the extended moving average acts as a low-pass filter.

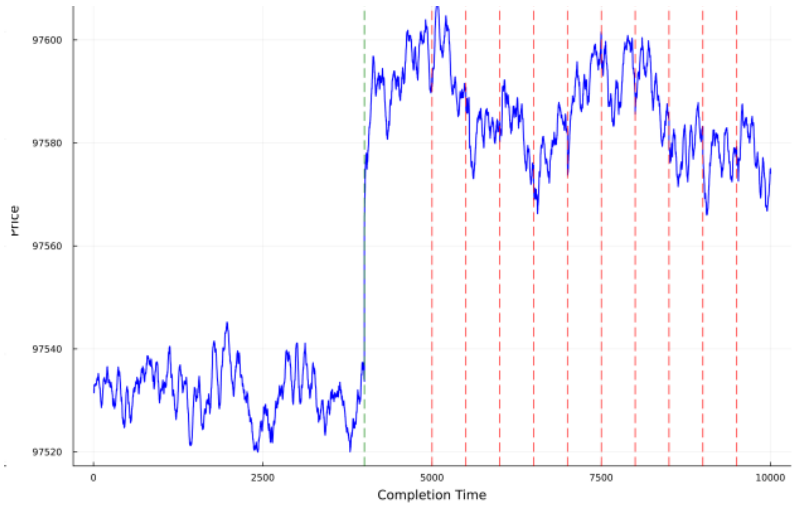


Figure 5: Price trajectory in the fragility pocket (N=50, ρ_{SL}=0.20). A sharp price discontinuity (green dashed line) is followed by repeated stop-loss cascade events (red dashed lines), illustrating the persistent instability that short agent memory and moderate stop-loss density produce.

Table 5: Comparative impact of stop-loss mechanisms on market stability.

Strategy	ρ_{SL}	**Kurt.**	**Vol.**	**Mean Err.**	J_{95}
No SL	0.00	13.79	0.00043	11.11	8.96
Strategic	0.01	8.16	0.00066	19.11	12.56
	0.05	5.16	0.00054	18.10	13.58
	0.10	4.15	0.00049	17.91	14.44
	0.25	2.41	0.00043	17.80	16.16
Uniform	0.01	5.57	0.00041	14.87	12.05
	0.05	4.29	0.00033	16.56	15.01
	0.10	4.75	0.00029	17.44	15.31
	0.25	5.83	0.00025	20.36	18.17

The Hurst exponent remains in the persistent regime ($H \approx 0.55$–0.65) across all simulations, including the volatile fragility pocket. Stop-loss cascades inject tail risk and volatility without destroying the memory structure of the price discovery process—matching empirical observations that real markets maintain long-range dependence even during high-volatility episodes.

5.4 Strategic vs. Uniform Stop-Losses

To distinguish the effects of agent intent from mechanical order presence, we compared two stop-loss mechanisms: a **Uniform** baseline (stop distances drawn uniformly between 0–1% of entry price) and a **Strategic** mechanism (trend-following agents use volatility-adaptive distances; stochastic agents use log-normal distances).

Table 5 presents the comparative results. Strategic stop-losses reduce excess kurtosis from a baseline of 13.79 (no SL) to 2.41 at $\rho_{\text{SL}} = 0.25$, while Uniform stop-losses plateau at approximately 5.83. This confirms that endogenous, entry-price-based risk management is more effective at diffusing cascades than exogenous random constraints.

A notable observation is the *Volatility Paradox*: strategic stop-loss adoption initially *amplifies* volatility at low adoption rates ($\rho_{\text{SL}} \approx 0.01$–$0.05$) before suppressing it. This occurs because at low concentrations, profit-taking orders actively collide with momentum flow, creating whipsaw dynamics. While this increases realised volatility, it prevents the formation of runaway drifts.

The rising error is expected: as stop-loss adoption increases, the simulated market becomes less fragile than the empirical BTC/USDT data it is calibrated against.

Table 6: Comparative stylized facts (CHAD vs. ABIDES, 219 runs).

Fact	Sim.	1s	5s	30s	60s
Kurtosis	CHAD	2.70	2.14	1.29	1.07
	ABIDES	159.59	28.12	3.99	1.41
Skewness	CHAD	−0.01	−0.03	−0.09	−0.09
	ABIDES	−0.46	−0.04	0.03	0.06
Leverage	CHAD	−0.003	−0.005	−0.029	−0.019
	ABIDES	−0.0003	−0.005	−0.001	−0.007

Table 7: Scaling performance (time in seconds).

	CHAD		ABIDES	
Agents	**Mean (±1SD)**	**Max**	**Mean (±1SD)**	**Max**
1,000	0.10 (0.08, 0.12)	0.24	23.39 (21.87, 24.90)	27.00
10,000	1.22 (1.07, 1.38)	1.58	53.96 (45.87, 62.04)	79.54
20,000	2.63 (2.25, 3.01)	7.13	101.26 (83.77, 118.75)	134.07
100,000	15.16 (13.79, 16.52)	16.52	Did Not Complete	
1,000,000	199.53 (181.14, 217.92)	233.02	Did Not Complete	

6 Comparative Benchmarks

6.1 Stylized Facts Across Time Scales

We benchmarked CHAD against ABIDES using identical experimental setups (the `rmsc03.py` configuration, one-hour trading session, 219 independent trials). Table 6 compares selected stylized facts across time scales.

CHAD's excess kurtosis exhibits a monotonic decay from 2.70 at 1-second to 1.07 at 60-second aggregation, consistent with empirical findings where leptokurtosis is most pronounced at high frequencies. ABIDES produces substantially higher kurtosis at 1-second (159.59) that decays sharply. CHAD also generates a weak but consistent negative leverage effect across all time scales (ranging from -0.003 to -0.029), while no consistent effect was observed for ABIDES under this configuration. We note that these leverage values are small in magnitude and that the 95% confidence intervals at higher frequencies approach zero; the effect is more robust at lower frequencies (30s and 60s).

6.2 Computational Performance

All benchmarks were conducted on a single machine (Intel i7-11700, 16 cores, 32 GB RAM, Ubuntu 20.04). CHAD was executed using Julia v1.11.6 [2]; ABIDES used Python v3.7.16.

On the `rmsc03` configuration, CHAD completed the one-hour simulation in 0.52 ± 0.20 s, compared to 68.98 ± 2.41 s for ABIDES (Table 7). This comparison requires qualification. The raw throughput difference reflects *two confounded factors*: (i) the architectural difference—CHAD's pre-scheduled design eliminates the message-passing kernel and dynamic event queue that ABIDES maintains to support asynchronous agent communication and network latency modelling; and (ii) the language-level advantage of Julia's JIT compilation over Python's interpreted execution. Isolating these contributions would require reimplementing one system in the other's language, which we have not done.

The more meaningful result is the scaling behaviour. Table 7 shows that CHAD maintains near-linear scaling from 1,000 to 1,000,000 agents, while ABIDES experienced run failures at the 100,000-agent scale. This difference is primarily architectural: CHAD's pre-computed schedule avoids the super-linear growth in queue-management overhead that event-driven systems incur as agent counts increase. It is this scalability—rather than raw speed on small configurations—that enables the large-scale ecological sweeps presented in Section 5.

7 Discussion and Open Questions

Several open questions arise from these results.

When are explicit agent strategies redundant? In our experiments, mean-reverting agents were statistically insignificant ($p = 0.44$) for reproducing stylized facts, with counter-trend pressure apparently supplied by the LOB structure via the limit orders of stochastic agents. This echoes the classic result of Gode and Sunder [9] that market mechanisms can partially substitute for individual rationality. While this finding is specific to the tested ecologies (which are dominated by stochastic liquidity providers), it raises a broader question for MABS: under what conditions does the structure of the interaction mechanism make explicit behavioural strategies redundant? Investigating this boundary between emergent and designed dynamics—across different agent compositions and domains—could inform more parsimonious agent design.

Calibration without ground truth. The validity corridor methodology exploits the availability of well-characterised stylized facts as ground truth. Many MABS domains (e.g., urban planning, epidemiology) lack such quantitative benchmarks. Recent work on surrogate-assisted calibration [12] offers promising approaches, but the fundamental question remains: how should agent ecologies be calibrated when the target emergent phenomena are qualitative or poorly characterised?

Timescale synchronisation and fragility. The "fragility pocket" at short agent time horizons ($N = 25$–50, $\rho_{\mathrm{SL}} \approx 0.10$–$0.20$) arises from a synchronisation between agent reaction times and cascade frequencies. Similar synchronisation effects may appear in other multi-agent systems where agents respond to collective outcomes that are themselves influenced by agent responses (e.g., traffic signal adaptation, supply chain coordination).

8 Conclusion

We have presented CHAD, an agent-based simulator that combines a pre-scheduled turn-based architecture with a dual-book system to model complex heterogeneous agent dynamics. Validation against empirical BTC/USDT data and a sensitivity analysis across 784 population configurations show that—within the tested parameter regime—macroscopic market properties such as fat tails, volatility clustering, and cascading liquidity events are primarily driven by agent population composition. The identification of distinct liquidity regimes, the statistical irrelevance of explicit mean-reversion in the tested ecologies, and a fragility pocket driven by timescale synchronisation illustrate how micro-level agent interactions give rise to macro-level phenomena. The architectural pattern and ecological analysis framework are not specific to financial markets and may generalise to other MABS domains.

References

[1] Belcak, P., Calliess, J.-P., Zohren, S.: Fast agent-based simulation framework with applications to reinforcement learning and the study of trading latency effects. arXiv preprint arXiv:2008.07871 (2020)

[2] Bezanson, J., Edelman, A., Karpinski, S., Shah, V.B.: Julia: A fresh approach to numerical computing. SIAM Review **59**(1), 65–98 (2017)

[3] Byrd, D., Hybinette, M., Balch, T.H.: ABIDES: Towards high-fidelity multi-agent market simulation. In: Proc. ACM SIGSIM Conf. on Principles of Advanced Discrete Simulation, pp. 11–22 (2020)

[4] Cliff, D.: BSE: A minimal simulation of a limit-order-book stock exchange. arXiv preprint arXiv:1809.06027 (2018)

[5] Cont, R.: Empirical properties of asset returns: stylized facts and statistical issues. Quantitative Finance **1**(2), 223–236 (2001)

[6] Cont, R., Stoikov, S., Talreja, R.: A stochastic model for order book dynamics. Operations Research **58**(3), 549–563 (2010)

[7] Farmer, J.D., Patelli, P., Zovko, I.I.: The predictive power of zero intelligence in financial markets. Proc. National Academy of Sciences **102**(6), 2254–2259 (2005)

[8] Franke, R., Westerhoff, F.: Structural stochastic volatility in asset pricing dynamics: Estimation and model contest. Journal of Economic Dynamics and Control **36**(8), 1193–1211 (2012)

[9] Gode, D.K., Sunder, S.: Allocative efficiency of markets with zero-intelligence traders: Market as a partial substitute for individual rationality. Journal of Political Economy **101**(1), 119–137 (1993)

[10] Gould, M.D., Porter, M.A., Williams, S., McDonald, M., Fenn, D.J., Howison, S.D.: Limit order books. Quantitative Finance **13**(11), 1709–1748 (2013)

[11] Kirilenko, A., Kyle, A.S., Samadi, M., Tuzun, T.: The flash crash: High-frequency trading in an electronic market. Journal of Finance **72**(3), 967–998 (2017)

[12] Lamperti, F., Roventini, A., Sani, A.: Agent-based model calibration using machine learning surrogates. Journal of Economic Dynamics and Control **90**, 366–389 (2018)

[13] Liston, P., Gretton, C., Lenskiy, A.: The role of stop-loss orders in market efficiency and stability: An agent-based study. In: Proc. 16th Int. Conf. on Agents and Artificial Intelligence (ICAART), pp. 280–288. SciTePress (2024)

[14] Lux, T., Marchesi, M.: Scaling and criticality in a stochastic multi-agent model of a financial market. Nature **397**(6719), 498–500 (1999)

[15] Mandelbrot, B.: The variation of certain speculative prices. The Journal of Business **36**(4), 394–419 (1963)

[16] Osler, C.L.: Stop-loss orders and price cascades in currency markets. Journal of International Money and Finance **24**(2), 219–241 (2005)

[17] Raberto, M., Cincotti, S., Focardi, S.M., Marchesi, M.: Agent-based simulation of a financial market. Physica A **299**(1–2), 319–327 (2001)