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

Patrick Liston[1] Charles Gretton[1] Artem Lensky[2]

[1]School of Computing, Australian National University, Canberra, ACT, Australia
[2]University of New South Wales, Canberra, ACT, Australia
Code: https://github.com/patrick-liston/CHAD.

Abstract

We introduce CHAD (Complex Heterogeneous Agent Dynamics) a high-performance agent-based financial market simulator written in Julia, designed to generate a realistic market microstructure and reproduce key *stylised facts* entirely through endogenous agent interaction.

CHAD introduces three principal design contributions: *(i)* a fully endogenous market simulator that generates macroscopic dynamics without exogenous price guidance; *(ii)* a dual-book system comprising a public Limit Order Book and a private Stop-Loss Order Book (SLOB), enabling explicit modelling of latent liquidity and cascading liquidation dynamics; and *(iii)* a pre-scheduled, turn-based execution architecture that pre-computes all agent activation times, enabling deterministic execution and near-linear computational scaling to millions of agents.

Through a comprehensive multi-dimensional parameter sweep, we demonstrate the emergence of macroscopic market properties, including fat-tailed return distributions, volatility clustering, and rapid liquidity cascades. Validation against empirical BTC/USDT data confirms that CHAD reproduces a broad range of stylised facts. Computational benchmarks further show that the pre-scheduled architecture combined with Julia's JIT compilation enables both faster and more stable execution, with over one million agents possible, a scale at which event-driven frameworks such as ABIDES struggle to complete.

Collectively, these results establish CHAD as both a scalable simulation engine and a structural contribution to market microstructure modelling, creating, to our knowledge, the first fully endogenous simulator to consistently replicate key stylised facts, and to feature a SLOB framework which provides a novel mechanism for studying systemic risk, liquidation cascades, and hidden liquidity dynamics in fully synthetic financial environments.

Keywords: Agent-based modeling (ABM); Multi-agent systems; Market microstructure; Financial simulation; Liquidity cascades; Emergent dynamics; Limit Order Book

1 Introduction and Motivation

Simulating financial markets is crucial for academics analysing market microstructure, traders evaluating strategies [1], and regulators assessing policy changes [17]. Because empirical studies are inherently bound to observed historical conditions, they cannot reliably predict market responses to novel regulations, structural shocks, or unseen trading regimes [16]. While classical stochastic models—such as fractional Brownian motion—capture basic volatility profiles, they struggle to replicate the non-linearities, extreme events, and recursive feedback loops native to real order books [2]. Agent-Based Models (ABMs) aim to resolve this by generating macro-level phenomena strictly through the micro-level interactions of heterogeneous participants.

Consequently, modern market simulators are widely deployed to evaluate regulatory mechanisms like price variation limits [23], short-selling constraints [21], and circuit breakers [13]. The current paradigm is dominated by platforms like ABIDES [5] and MAXE [4], which build upon classical frameworks [19, 20] to show that adaptive agents can reproduce common statistical regularities, or *stylised facts* [14, 8].

While these established platforms like ABIDES [5], the Bristol Stock Exchange [7], and MAXE [4] have advanced the study of market microstructure, their event-driven architectures rely on dynamic priority queues and asynchronous scheduling. This introduces significant computational overhead, limiting scalability and complicating exact reproducibility at large agent counts.

Furthermore, existing simulators typically omit critical conditional execution mechanisms, notably stop-loss orders, which have been linked to liquidity cascades and market fragility [18, 15, 5]. Although some recent work has incorporated Limit Order Book (LOB) structures and induced stop-loss cascades thus replicating heavy-tailed distributions, the framework remained dependent on historical data anchoring and suffered from constrained computational scalability [15]. Crucially, many contemporary simulators require such historical data or exogenous price paths to maintain stability, sacrificing true endogeneity.

To investigate the systemic risk of increasingly speculative markets in which no true fundamental value exists, new agent-based simulators must be capable of generating realistic micro and macro-dynamics strictly as emergent properties of agent interaction.

While previously mentioned simulators often rely on historical inputs or mathematically defined guidance prices—through the use of fundamental value agents or some other guidance mechanism—we are concerned with completely endogenous simulation that does not rely on such inputs.

We introduce CHAD (Complex Heterogeneous Agent Dynamics), a high-performance, fully synthetic simulation framework implemented in Julia. CHAD delivers three core contributions that position it uniquely within the literature:

1. **Pure Endogenous Realism:** Equipped with a configurable, heterogeneous agent ecology, CHAD achieves realistic macroscopic dynamics without exogenous price guidance. We validate the framework against empirical BTC/USDT stylised facts as per [8] and further by using a validity corridor methodology. We also display that CHAD has the ability to natively trigger emergent flash crash scenarios.

2. **Mechanistic Completeness via Dual-Book Architecture:** CHAD formalizes a public LOB alongside a private Stop-Loss Order Book (SLOB). This dual-book framework captures hidden liquidity, can model recursive, path-dependent stop-loss cascades, and introduces the structural capacity to examine the impacts of latent liquidity, exploitative trading strategies, and phenomena such as "stop-loss hunting."

3. **Computational Scalability and Determinism:** Rather than relying on asynchronous event queues, CHAD implements a pre-scheduled, turn-based execution architecture. Precomputing agent activation times eliminates queue-management overhead, guaranteeing deterministic execution and enabling near-linear scaling to millions of autonomous agents. This results in enhanced speed as compared to previous systems.

2 CHAD Simulator Architecture and Order Execution

To optimise computational efficiency for large-scale market experiments, the CHAD framework departs from traditional discrete-event architectures that rely on dynamic event queues. Instead, it employs a pre-scheduled, turn-based design where a master schedule of all agent decision points is computed at initialisation, allowing the engine to sequentially advance through active time steps and skip periods of inactivity.

Figure 1 outlines the framework's structural hierarchy, split into three core layers: the central Market engine, autonomous Agents (along with their atomic Data Structures), and a dual-book system that segregates visible limit orders (Public LOB) from latent, trigger-based liquidity (Private SLOB).

At initialisation, a scheduling function τ computes the exact set of timestamps, $T = \{t_1, t_2, \ldots, t_n\}$, at which specific agents activate. When an agent is triggered at tick t_i, it undergoes a four-step lifecycle: **Perception** –querying market state variables such as the last executed price P_{t-1}^{last}, **Evaluation** –processing state updates via archetype-specific logic, **Execution Formulation** – calculating trade directions and volume sizes into an `Order` struct (if executing a trade), and **Submission** –sending the order to the core matching engine (if executing a trade).

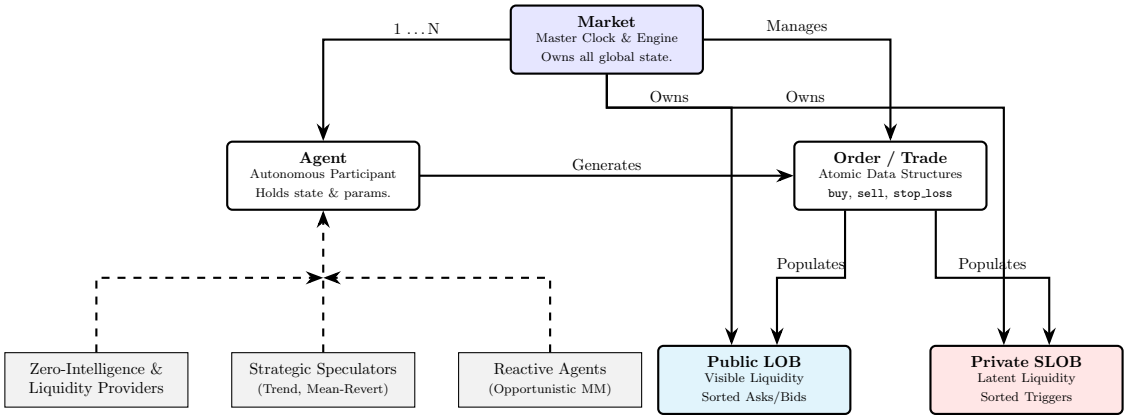


Figure 1: Structural architecture of the CHAD simulation framework.

2.1 Order Processing and Execution Mechanics

CHAD models a continuous double-auction market. When multiple agents share an identical execution timestamp t_i, a potential race condition arises. By default, the engine shuffles this activation array to process agents in a randomised sequence, preventing artificial structural precedence. Alternatively, the framework can sort this array dynamically (e.g., by agent ID or latency tier) to simulate asymmetric market access, such as institutional co-location advantages over retail participants.

The centralised exchange manages a public Limit Order Book (LOB), denoted $\mathcal{L}_t$, and a private Stop-Loss Order Book (SLOB), denoted $\mathcal{S}_t$. The best bid B_t and best ask A_t are defined as:

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

An individual order is defined by the tuple $O = (d, q, p)$, where $d \in \{\text{buy}, \text{sell}\}$, q is quantity, and p is the limit or trigger price. CHAD facilitates the placement of three distinct order types:

- **Market Orders** immediately cross the book against the opposing best available price levels. Unfilled residual quantities are automatically cancelled ($q_{\text{residual}} \to$ Cancelled).
- **Limit Orders** execute immediately if $p \geq A_t$ (for buys) or $p \leq B_t$ (for sells); otherwise, they rest transparently on the LOB ($\mathcal{L}_{t+1} = \mathcal{L}_t \cup \{O\}$).
- **Stop-Loss Orders** are kept private within $\mathcal{S}_t$. Their trigger conditions are evaluated continuously against the most recent transaction price P_t^{last}:

$$\text{Trigger}(O) = \begin{cases} \text{True} & \text{if } d = \text{sell} \wedge P_t^{\text{last}} \leq p_{\text{trig}} \\ \text{True} & \text{if } d = \text{buy} \wedge P_t^{\text{last}} \geq p_{\text{trig}} \end{cases} \tag{2}$$

  Upon activation, the order is transformed into a market order.

Liquidity Cascades and Recursive Execution Loops Orders must pass agent-side validation checking for sufficient capital and compliance with an open order limit ($N_{\max}$). Once validated, matched transactions update P_t^{last}, which instantly prompts an evaluation of the SLOB.

Because triggered stop-losses convert into market orders, their execution consumes resting liquidity and can establish new price bounds P_{t+k}^{last}. The engine therefore recursively re-evaluates remaining stop-loss orders at each newly generated micro-tick price level. This cascade continues until the strict termination condition is satisfied:

$$\forall O \in \mathcal{S}_{t+k}, \quad \text{Trigger}(O) = \text{False} \tag{3}$$

This loop completely resolves all cascading liquidity shifts within the current master tick before the clock increments to t_{i+1}, ensuring absolute determinism. (See Figure 2(b) for a visual depiction.)

Architectural Abstractions CHAD abstracts away nanosecond-level physical network transport delays to preserve high-throughput macro scalability. High-level strategic latency (such as market maker reaction windows) is instead handled natively via discrete simulation tick gaps assigned to specific agent life-cycles.

For a comprehensive operational overview of these mechanisms, Figure 2 maps the complete execution pipelines of the framework. Panel(a) details the pre-scheduled progression of the master simulation clock, while Panel(b) provides the step-by-step logic of an individual order life-cycle.

2.2 Temporal Mapping, Persistence, and Output Metrics

CHAD operates as a fully persistent simulator in which all market activity is recorded at the level of individual orders, trades, and latent liquidity states. To enable comparison with empirical financial time series, discrete simulation ticks $i = 1, \ldots, N_{\text{ticks}}$ are mapped onto a uniform temporal grid:

$$t_i = T_{\text{start}} + i \cdot \frac{T_{\text{end}} - T_{\text{start}}}{N_{\text{ticks}}}. \tag{4}$$

The number of ticks can be calibrated to empirical trading intensity such that the effective trade density matches observed market volume (e.g., BTC/USDT). This ensures that agent activity is embedded within a physically consistent temporal scaling regime, allowing direct comparison of simulated and historical time series.

A key feature of the framework is that all market states are fully reconstructable *ex post*. Every order submission, execution, and cancellation is stored in a complete audit trail, enabling the exact reconstruction of both the active limit order book and the latent stop-loss order book at any time t. This persistence property eliminates the need to re-execute the simulation engine to inspect or recreate historic market states for analysis and ensures reproducibility of all derived results.

From these raw event logs, all standard empirical representations can be generated, including OHLC candles, log returns, and higher-order statistical diagnostics. Table 1 summarises the core persistent data structures and the downstream objects derived from them.

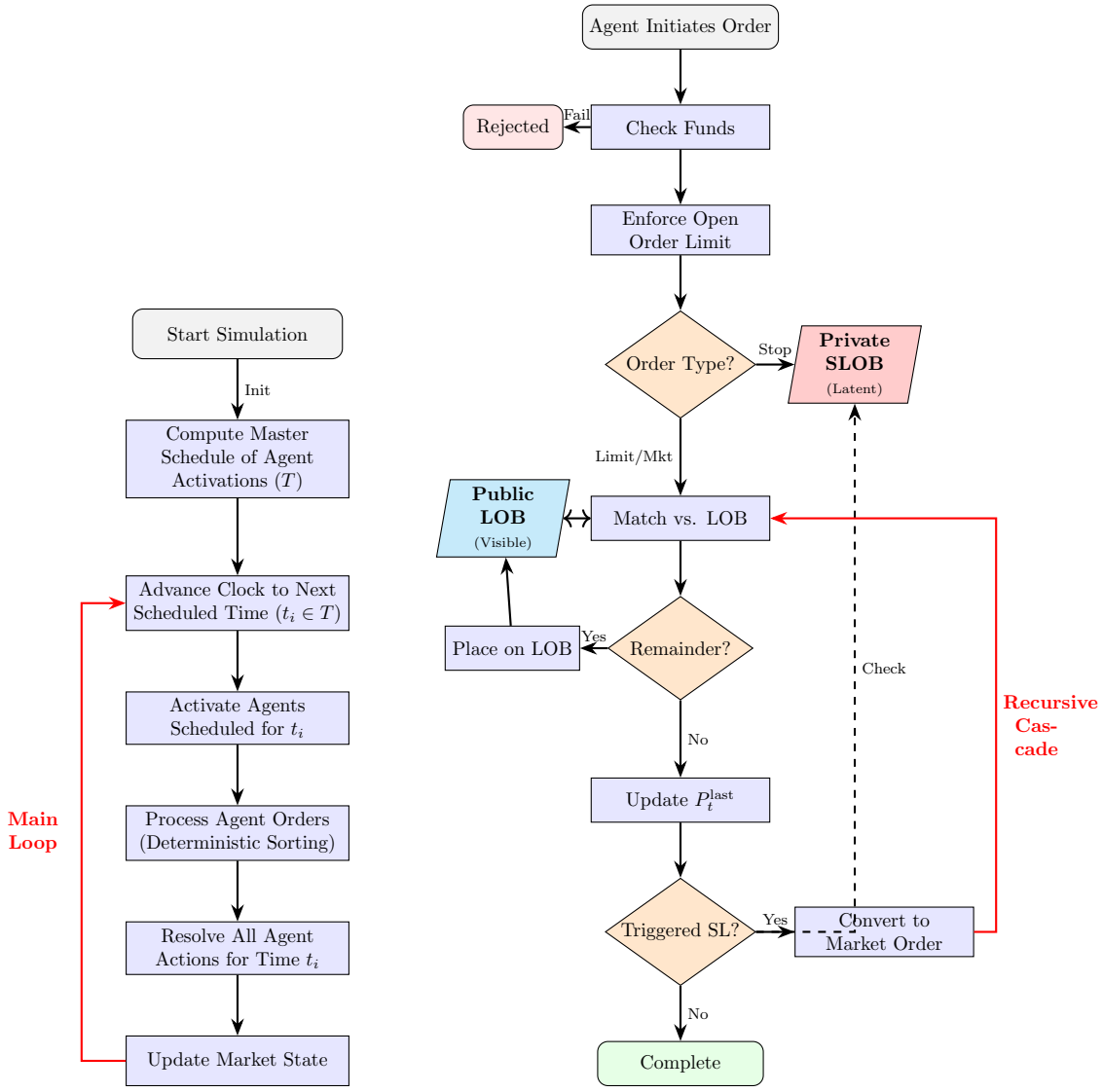
Table 1: Core simulation data structures and derived analytical outputs in CHAD.

Struct	**Key Fields**	**Role in Simulation and Analysis**
Agent	`agent_id`, `cash/shares` , `params`	Encodes agent decision rules and portfolio state; fully parameterised agent.
Order	`id`, `type`, `time`, `price`, `remaining`	Records all limit and stop orders; enables full reconstruction of order flow and LOB dynamics.
Trade	`trade_id`, `buyer/seller` `amount`, `price`, `aggressor_side`	Atomic execution record capturing matched transactions and market clearing events.
LOB	`asks/bids` , `sl_asks/sl_bids`	Dual-book structure containing active liquidity and latent stop-loss liquidity at each price level.
Market Logs	`all_orders`, `trades`, `price_history`	Complete audit trail enabling exact reconstruction of historical market states and time series.

This architecture ensures that all empirical analyses are derived from a single, consistent source of truth. Because both active and latent liquidity are persistently recorded, any observed stylised fact can be traced back to its microstructural origin. This provides a direct bridge between agent-level behaviour, order book dynamics, and aggregated statistical outputs.

3 Agent Architecture and Parametric State Space

Rather than hard-coding rigid classes, CHAD uses an `Agent` structure combining a fixed core of shared state variables (`agent_id`, `cash_holding`, `shares_holding`, `num_open_trades` – Table 1) with a dynamically extensible `params` dictionary. This architecture enforces a uniform ecosystem interface while allowing per-agent behavioural extensions. Crucially, because the `params` dictionary can be expanded on a per-agent basis, it easily supports advanced, state-dependent attributes



(a) Pre-scheduled execution architecture.

(b) Order lifecycle and execution loop.

Figure 2: Complete execution pipelines of the CHAD simulator architecture.

— such as unique execution pathways or machine learning and reinforcement learning weights — without breaking the common structural framework. To prevent artificial synchronisation, these autonomous agents are differentiated across multiple dimensions to generate a deeply heterogeneous population. Using the parametric distributions mapped in Table 2, each agent is assigned distinct traits, including strategy type, wake/trade frequency, trade size, look-back window (N), sensitivity, and stop-loss placement propensity.

Table 3 defines the parameter set used for all experiments covered in the following sections. 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.

Agent Archetypes CHAD supports a modular set of agent archetypes, it allows for extensibility and the inclusion of user preferred agents. For our experiments we use a small set of relatively simple agents common within the literature [5, 11, 4]. Let p_{t-1}^{last} denote the last observed market price, p_t^{oracle} a fundamental price from an external series, and $\text{SMA}_N(t) = \frac{1}{N}\sum_{i=1}^{N} p_{t-i}^{\text{last}}$ the simple

Table 2: Core Attributes and Strategic Parameters of the `Agent` Structure

Attribute / Parameter	**Type**	**Description**
State and Identity Metrics		
`agent_id`	Integer	Unique identifier for deterministic execution and tracking.
`cash_holding`	Float	Current fiat or quote currency balances.
`shares_holding`	Float	Current asset or base currency holdings.
`num_open_trades`	Integer	Active order count used for open order limits.
Strategic and Behavioural Parameters		
`:agent_type`	Symbol	Defines underlying trading logic (e.g., `:Trend_Following`).
`:sample_freq`	Integer	Decision frequency interval
`:trade_size`	Float	Defines order sizing parameters (fixed, fractional, etc.).
`:N`	Integer	Look-back window size (temporal memory).
`:sensitivity`	Float	Archetype-specific activation threshold.
`:max_open`	Integer	Maximum open order allocations permitted.
`:stop_loss_prob`	Float (ρ_{SL})	Probability of an agent to set a stop-loss.

Table 3: 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	$\mathcal{U}(0.0, 0.3)$

moving average over look-back period N. The archetypes are:

Heuristic Stochastic Agents provide random liquidity in the tradition of zero-intelligence traders [11, 9], they extend a stochastic model with a simple rule to avoid immediate losses, constraining the order price to be advantageous relative to the current market price. These participants augment the baseline stochastic model by implementing a passive spread-capturing heuristic to limit immediate adverse execution:

$$p_t^{\text{order}} = \begin{cases} P_{t-1}^{\text{last}} \cdot (1-\epsilon) & \text{if } d_t = \text{buy} \\ P_{t-1}^{\text{last}} \cdot (1+\epsilon) & \text{if } d_t = \text{sell} \end{cases} \qquad \epsilon \sim \mathcal{U}(0, \epsilon_{\max})$$

Trend-Following Agents: Implementing a momentum configuration, these agents track a simple moving average ($\text{SMA}_N(t) = \frac{1}{N}\sum_{i=1}^{N} P_{t-i}^{\text{last}}$) over look-back horizon N, executing market orders in the direction of the macro trend:

$$d_t = \begin{cases} \text{buy} & \text{if } P_{t-1}^{\text{last}} > \text{SMA}_N(t) \\ \text{sell} & \text{if } P_{t-1}^{\text{last}} < \text{SMA}_N(t) \end{cases}$$

Mean-Reverting Agents: Operating on a contrarian premise to trend-followers, these participants assume short-term price innovations are transient. Consider the inverse of the formulas laid out for trend-following agents.

Market Maker Agents: Provide continuous two-sided liquidity based on the framework in [3]. They continuously adjust their reservation price r as a function of their current inventory asset position q, risk aversion parameter γ, and estimated volatility σ:

$$r(s, q, t) = P_t^{\text{mid}} - q\gamma\sigma^2 \tag{5}$$

They iteratively quote bid and ask limits relative to an endogenous half-spread configuration $\delta(s, q, t)$, dynamically managing inventory accumulation risk.

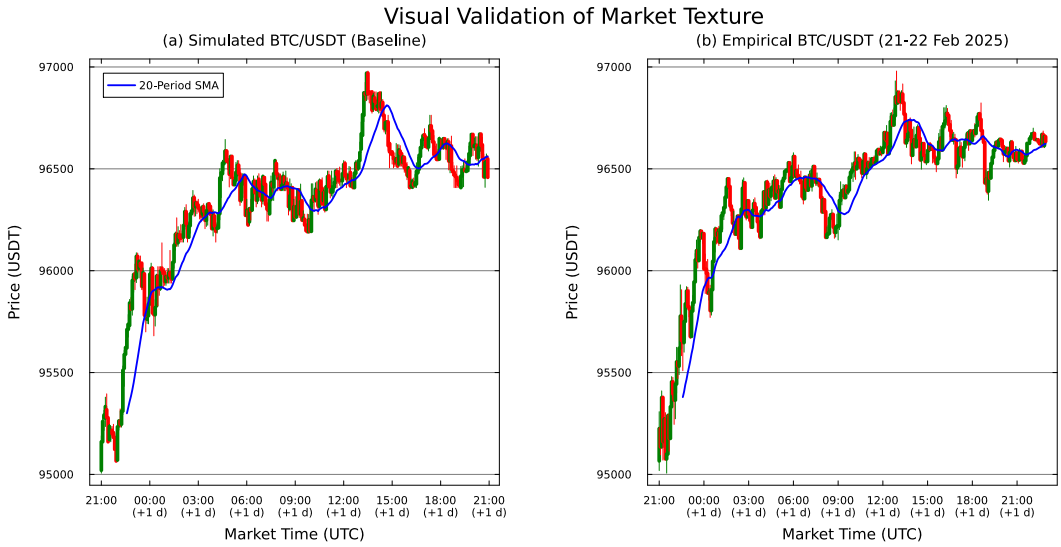


Figure 3: Comparison of a simulated price trajectory against historical BTC/USDT data.

Value Agents: Relying on an exogenous fundamental value process P_t^{oracle}, these agents trade mean-reverting deviations against a perceived asset Value agents are integrated strictly to benchmark execution profiles against legacy simulators (i.e., ABIDES [5]) and are disabled during purely endogenous experiments.

Trade Size Determination 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 can closely replicate empirical BTC/USDT trade size distributions for our given study period. However, for our studies in this paper we sample (without replacement) from historical tick-level BTCUSDT trade quantities found on the Binance exchange during the first 14 days of November 2021.

Agent Initialisation The global agent ecology configuration is defined prior to execution. To isolate strategic behavioural mechanics from funding constraints and avoid early bankruptcy artifacts, initial wealth endowments are initialized uniformly across all core experimental setups at cash $= 1.0 \times 10^{10}$ and base asset $= 1.0 \times 10^{9}$. This configuration relaxes binding capital constraints, though the system retains the flexibility to process constrained or asymmetric initial wealth distributions if desired.

4 Validation & Sensitivity Analysis

The validity of an agent-based model is demonstrated by its capacity to endogenously reproduce statistical properties observed in real-world financial data. These empirical properties, known as *stylised facts*, serve as established benchmarks for assessing how realistically a simulator captures market behaviour.

This section outlines the validation of the CHAD simulator through a sequential process: first, we establish a baseline agent ecology to evaluate time-series stylised facts and Limit Order Book (LOB) microstructure against empirical crypto-asset data. Second, we introduce a sensitivity analysis and calibration framework utilising a rolling *Validity Corridor* to account for market non-stationarity. Finally, we demonstrate the model's tunability via a Method of Simulated Moments (MSM) error reduction protocol.

Prior to more rigorous mathematical and statistical tests, however, we present a qualitative presentation of the output, as compared to the real price path, Figure 3.

Despite the simulation being initialised only with a single order book snapshot and evolving through purely synthetic interactions thereafter, the simulation (as shown in Figure 3) almost perfectly mirrors the price path that occurred in reality. This is likely an uncommon occurrence, but shows the simulators capability to endogenously produce results that accurately capture the dynamics of real markets, and further supports our claim that we may adequately model them through this simplified view without the need to exogenous guidance that previous simulators have relied on. Given the simulator's capacity to replicate realistic market texture and price direction movement from a single snapshot, the path similarity does also suggest the LOB likely plays dominant role in on short-term price formation.

4.1 Baseline Validation: Time-Series Stylised Facts

Pursuant to the validation frameworks established in [8, 6], we evaluate the simulator's generative capabilities under a initial *Baseline Agent Ecology*. The objective of this preliminary analysis is to confirm that the CHAD simulator endogenously produces the fundamental non-linear statistical signatures of financial time-series before subjecting the model to the granular, product-specific constraints of the BTC/USDT validity corridor.

Our baseline validation focuses on log-returns, calculated as $r_t = \ln(P_t/P_{t-1})$, across 100 independent Monte Carlo simulation runs using a fixed configuration intended to represent a liquidity-rich market environment. This baseline population consists of $N = 6{,}100$ agents, with the composition detailed in Table 4.

Table 4: Baseline Agent Population Composition

Agent Type	**Count**
`:HeuristicStochasticTraders`	5,000
`:MarketMakers`	100
`:TrendFollowers`	500
`:MeanReverters`	500

Table 5: Statistical Summary of Time-Series stylised Facts (Baseline Ecology, 100 Runs)

Statistic	Mean	Std. Dev.	**95% CI for Mean**	**Common Range**
Excess Kurtosis	5.9012	0.1076	[5.8798, 5.9226]	4–10+ [8]
Skewness	−0.0126	0.0202	[-0.0167, -0.0086]	-0.5 to 0.1 [8]
Leverage Effect*	−0.0002	0.0037	[-0.001, 0.0005]	-0.1 to -0.6 [6]

This table presents aggregated statistics for the baseline simulation. *$\mathrm{Corr}(r_t, \sigma^2_{t+\tau})$

Importantly, the "common ranges" in Table 5 are derived from equity and commodity literature, and do not fully account for the higher variation typical of cryptocurrency markets. This divergence is explored later in this section and motivated our development of the *validity corridor*. Consequently, while the baseline ecology successfully reproduces the defining "fat-tailed" distribution of market returns, the secondary volatility dynamics require further tuning:

Leptokurtic Distributions (Fat Tails): The simulated returns exhibit a mean excess kurtosis of 5.90. This confirms that the simulator generates extreme price movements ("black-swan" events) significantly more frequently than a Gaussian random walk, aligning with empirical ranges reported in [8]. **Statistical Asymmetry:** The mean skewness is slightly negative (-0.0126), consistent with the empirical observation that large negative returns (crashes) are statistically more probable than equivalent positive surges. The simulator captures this asymmetry organically from agent interactions without exogenous forcing. **Absence of Leverage Effect:** In this specific baseline configuration, the correlation between past returns and future volatility is effectively zero (-0.0002). While a negative correlation (the "Leverage Effect") is typical in equity markets, its absence here indicates that the volatility-feedback loop is weak in the current agent mix. This discrepancy highlights the necessity of the sensitivity analysis presented in Section 4.3.

4.2 Limit Order Book Microstructure Validation

While time-series metrics capture the macroscopic outcomes of price formation, the fidelity of the CHAD simulator must also be evaluated at the microstructural level. To this end, we benchmark the simulated Limit Order Book (LOB) against a high-resolution empirical baseline reconstructed from historical BTC/USDT data using a *"Snapshot and Replay"* methodology [12].

The reconstruction process initiates with a Level-2 snapshot (t_0) comprising the top 500 price levels on both the bid and ask sides. We subsequently apply the full sequence of tick-level order updates (insertions, cancellations, and executions) over time. This approach allows for the accumulation of deep liquidity beyond the initial snapshot horizon, ensuring that the reconstructed book reflects true latent supply and demand curves.

To mitigate the effects of intraday autocorrelation, we employ a *Stratified Regime Sampling* strategy, selecting 96 discrete snapshots distributed across three temporally disjoint trading days (08-11-2024, 30-11-2024, and 22-02-2025). This ensures regime diversity across varying volatility bounds and guarantees temporal decorrelation, as the intervals significantly exceed the typical relaxation time of the LOB. Table 6 compares the structural characteristics of the simulated LOB against this empirical ground truth.

Table 6: Microstructural Comparison: Simulated vs. Empirical Limit Order Book Statistics

	Simulated (Baseline)	**Empirical (BTC/USDT)**
Metric	**Mean [95% CI]**	**Mean [95% CI]**
Bid–Ask Spread	0.1108 [0.09, 0.12]	0.1021 [0.09, 0.11]
Total Bid Amount	107.60 [106.9, 108.2]	152.43 [150.7, 154.2]
Total Ask Amount	177.85 [173.9, 181.8]	171.31 [168.4, 174.3]
LOB Imbalance	-0.2430 [-0.25, -0.23]	-0.0483 [-0.05, -0.04]

This table compares the average state of the limit order book in the simulation against empirical BTC/USDT data. The LOB imbalance is defined as $(Q_{\text{bid}} - Q_{\text{ask}})/(Q_{\text{bid}} + Q_{\text{ask}})$.

The comparative analysis yields two key insights regarding microstructural fidelity; **Spread Precision:** The simulated Bid-Ask spread (0.1108) aligns closely with the empirical mean (0.1021). This suggests that the interaction mechanics between `:MarketMaker` agents and liquidity takers are accurately calibrated to provide the requisite level of market liquidity. **Depth Asymmetry:** While the *Total Ask Amount* matches reality well, the simulation exhibits a deficit in *Total Bid Amount* (107.6 vs 152.4), inducing a stronger negative imbalance (-0.243) than the near-neutral empirical market (-0.048). This structural selling pressure highlights the need for the parameter tuning framework introduced below.

4.3 Sensitivity Analysis and Calibration Framework

To verify that the simulator does not over-fit to a single regime, we establish a framework to demonstrate its **tunability**—the capacity to traverse a wide range of market behaviours by systematically varying the composition of its agent ecology.

Statistical Measures and Stylised Facts To evaluate the statistical fidelity of the simulated market dynamics, we compute a standard set of stylised facts widely documented in the financial econometrics literature [8]. Logarithmic returns where the return is defined as $r_t = \ln(P_t) - \ln(P_{t-1})$, and was captured on 1-minute aggregated open price basis.

Following established practices, we assess distributional, dependence, and memory properties using volatility (σ), tail heaviness via the Hill tail index, return and absolute-return autocorrelation functions, the Hurst exponent, and GARCH persistence. Together, these metrics capture the key empirical regularities of financial time series, including heavy tails, volatility clustering, and long-range dependence.

Empirical Grounding: The Validity Corridor To address the non-stationarity inherent in crypto-asset data, we decompose a high-frequency BTC/USDT dataset (2017–2024) into rolling 24-hour windows. This builds a non-parametric distribution of daily stylised facts, defining our *Validity Corridor*.

The Validity Corridor – A simulation run S is considered empirically valid with respect to a stylised fact F if the scalar value $F(S)$ falls within the 5th and 95th percentiles of the empirical daily distribution of F observed in the reference market.

Table 7: Empirical Validity Corridor (BTC/USDT Daily Bounds)

Metric	**Global Mean**	**Daily Mean**	**5% Bound**	**95% Bound**
Distributional Properties				
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
Temporal Dependence & Memory				
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
Leverage Eff.	0.0309	0.0283	−0.1321	0.2063

Vol Clustering is measured by the ACF of absolute returns, $|r|$, at lag 1. **Return ACF** is the ACF of raw returns at lag 1. Global aggregates calculated over the full 2017–2024 horizon.

The stark divergence between Global and Daily statistics in Table 7 demonstrates why traditional long-run averaging is inadequate for highly volatile markets (i.e cryptocurrency); global metrics are dominated by rare, extreme structural shifts that mask typical daily trading conditions. Thus we aim to fall within a two-sigma "validity band" in order to satisfy the key stylised fact conditions.

The Objective Function: Method of Simulated Moments To systematically tune the simulator's configuration, we define a macroscopic parameter vector θ representing the agent ecology mix:

$$\theta = [N_{\text{Stoch}}, N_{\text{Trend}}, N_{\text{MeanRev}}, N_{\text{MM}}] \tag{6}$$

where each element maps to the total population count of a primary agent class.

To score these configurations, we implement a heuristic Method of Simulated Moments (MSM) (inspired by the Method of Simulated Moments (MSM) [10, 22]). With a loss function $J(\theta)$ utilising a diagonal weighting scheme based on absolute z-score deviations:

$$J(\theta) = \sum_{i=1}^{K} w_i \left| \frac{m_{\text{sim},i}(\theta) - \mu_{\text{data},i}}{\sigma_{\text{data},i}} \right| \tag{7}$$

where $m_{\text{sim},i}(\theta)$ is the simulated moment for metric i, $\mu_{\text{data},i}$ is the empirical validation mean, and $\sigma_{\text{data},i}$ is the empirical standard deviation. This normalisation acts as a stabiliser, ensuring high-magnitude metrics (e.g., Kurtosis) do not disproportionately dominate low-magnitude parameters (e.g., Volatility). The loss function aggregates $K = 11$ moments covering our distributional, temporal dependence, and autocorrelation lag structures.

Weighting Scheme and Protocol We assign double weight ($w_i = 2.0$) to Kurtosis and the Hurst Exponent to prioritise the non-linear hallmarks of crypto-asset markets (heavy tails and long memory). Other metrics carry a unitary weight ($w_i = 1.0$), while the Hill Index and Leverage Effect are excluded from explicit tuning to safeguard from convergence against sampling noise.

To map the error surface generated by $J(\theta)$, we deploy a hypercube combinatorial grid search generated via the Cartesian product of the following bounded allocations:

- **Stochastic Agents:** $N_{\text{Stoch}} \in \{1, 500, 1000, 2000, 3000, 3500, 4000, 5000\}$
- **Trend Followers:** $N_{\text{Trend}} \in \{0, 100, 250, 500, 750, 1000\}$
- **Mean Reverters:** $N_{\text{MeanRev}} \in \{0, 100, 250, 500, 750, 1000\}$
- **Market Makers:** $N_{\text{MM}} \in \{10, 50, 100, 200\}$

Results: Sensitivity Landscape and Calibration We map the topology of the error surface $J(\theta)$ to identify where the simulated microstructure aligns with empirical stylised facts. The dominant factor driving model fidelity is the population of stochastic liquidity providers (N_{Stoch}). Figure 4 illustrates this relationship across the simulated ecologies.

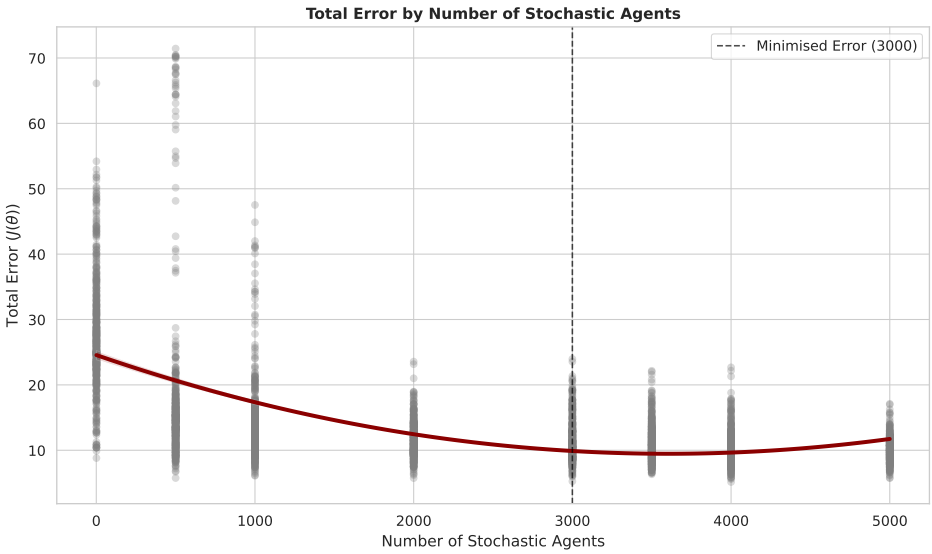


Figure 4: **Sensitivity of Aggregate Error $J(\theta)$ to Market Liquidity.** The black line represents the mean quadratic trend across independent simulation runs (grey dots). A stable error region emerges near $N_{\text{Stoch}} \approx 3{,}000$, demonstrating bounded error behaviour.

The error surface reveals a progression across three microstructural regimes:

Liquidity Crisis ($N_{\text{Stoch}} < 1{,}000$): Brittle market depth fails to absorb order flow, causing fragmented price discovery, severe inter-run variance, and high errors ($J(\theta) > 40$). **The Plausible Zone ($1{,}000 \leq N_{\text{Stoch}} \leq 3{,}500$):** Error decays rapidly. The book achieves the structural depth required to smoothly intermediate trading, allowing volatility clustering and fat tails to emerge endogenously. **Stabilisation ($N_{\text{Stoch}} \geq 4{,}000$):** Further depth yields diminishing returns. Excessive liquidity can over-dampen natural noise, rendering the price process unrealistically Gaussian compared to crypto-asset benchmarks.

A key operational strength of the CHAD framework is its structural stability. While heavily influenced by foundational liquidity, the aggregate error $J(\theta)$ remains remarkably bounded and consistent across diverse agent permutations. Once the baseline liquidity threshold ($N_{\text{Stoch}} \geq 2{,}000$) is satisfied, the model avoids catastrophic structural collapse or hyper-sensitivity to minor variations in speculative volume. To decouple these dynamics, we regressed the normalised agent population z-scores against total error $J(\theta)$.

Ultimately, this robust error landscape supports the notion that the chosen calibration zone ($N_{\text{Stoch}} = 3{,}000$) is driven by predictable, balanced macro-dynamics rather than brittle parametric configurations.

Table 8: Comparative stylised 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

5 Comparative Benchmarks

Stylised Facts Across Time Scales We benchmarked CHAD against ABIDES using identical experimental setups. We utilised the `rmsc03.py` [5] configuration for ABIDES and mirrored this architecture within CHAD, implementing a one-hour trading session across 219 independent trials. Table 8 compares selected stylised facts across varying temporal scales.

The excess kurtosis generated by CHAD exhibits a monotonic decay from 2.70 at a 1-second aggregation to 1.07 at a 60-second aggregation, consistent with empirical findings where leptokurtosis is most pronounced at ultra-high frequencies. Conversely, ABIDES produces substantially higher excess kurtosis at the 1-second interval (159.59), which subsequently decays sharply. Furthermore, CHAD generates a weak but consistent negative leverage effect across all temporal scales (ranging from -0.003 to -0.029), whereas no stable trend is observed for ABIDES under this specific configuration. While these leverage values are small in magnitude and their 95% confidence intervals approach zero at higher frequencies, the effect becomes notably more robust at lower frequencies (30s and 60s). Although ABIDES may not have been natively designed to reproduce statistical stylised facts at such granular intervals, the divergence in distributional stability is noteworthy; CHAD demonstrates significantly more robust and stable statistical properties across the evaluated time frequencies.

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; 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 9). 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 re-implementing one system in the other's language, which we have not done.

The more meaningful result is the scaling behaviour. Table 9 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 4.3.

6 Stop-Loss Mechanism and Microstructural Validation

Building on the validated CHAD baseline, we introduce a further structural extension: the Stop-Loss Order Book (SLOB). This section evaluates alternative stop-loss placement mechanisms, motivates our adopted specification, and verifies that the resulting microstructure remains consistent with empirical market behaviour.

Table 9: 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	

Stop-Loss Placement Mechanisms Because real-world stop-loss distributions are fundamentally unobservable, any implementation must be treated as a stylised behavioural approximation rather than a direct empirical reconstruction. To isolate the role of latent liquidity structure, we model stop-loss attachment as a probabilistic extension of executed trades.

Following an entry order O_{entry} at price p_t, a Bernoulli trial determines whether a stop-loss is placed:

$$\mathbb{I}_{\text{place}} = \begin{cases} 1 & \text{if } X < \rho_{\text{SL}} \\ 0 & \text{otherwise} \end{cases}, \quad X \sim \mathcal{U}(0,1) \tag{8}$$

If placed, a conditional order is inserted into the SLOB with opposite directionality to the original order. Depending on the agent cohort, we evaluate three placement methods:

1. **Uniform Placement (Baseline):** Stop distances are sampled uniformly and independently of agent state ($\delta \sim \mathcal{U}(0, 0.01)$), producing a diffuse, homogeneous liquidity layer.

2. **Volatility-Adaptive Placement:** Stop distances d_t scale dynamically with recent market volatility:
$$d_t = \max\left(0.01, \frac{1.5 \cdot p_t \cdot \sigma_\tau}{10^6}\right)$$
where σ_τ is the standard deviation of log returns over a rolling τ-tick window (defaulting to a static 1% if history is insufficient). This mimics a dynamic Bollinger Band to prevent premature liquidations during routine noise.

3. **Stochastic Placement:** Stop distances follow $\delta \sim \text{LogNormal}(-4.91, 0.8)$, a heavy-tailed distribution, clamped to $[0.002\%, 10\%]$. This parameterisation was calibrated via preliminary sensitivity testing.

Market Impact of Latent Liquidity To evaluate the effect of placement structure, we simulate market dynamics under no stop-losses, uniform placement, and heterogeneous (adaptive/stochastic) configurations.

Uniform placement acts primarily as a stabilising mechanism, diffusing latent liquidity and dampening extreme price movements. In contrast, heterogeneous placement concentrates stop orders spatially, enabling local liquidity depletion and endogenous cascade dynamics. Across stylised-fact validation metrics, the heterogeneous specifications consistently produce realistic volatility clustering, heavy-tailed returns, and episodic liquidity shocks, while the uniform model systematically underestimates tail risk. These results indicate that the spatial distribution of stop-loss orders—not merely their presence—is the critical determinant of market impact.

Accordingly, we adopt the stochastic method for all subsequent experiments. We omit the corresponding distribution plots here for space considerations, though interested readers are encouraged to review our other works for more detail.

Microstructural Validation To verify that the dual-book architecture is internally consistent, we examine the joint evolution of the public LOB and latent SLOB, as displayed in Figure 5. Qualitatively, several key microstructural properties emerge from the spatiotemporal liquidity distribution:

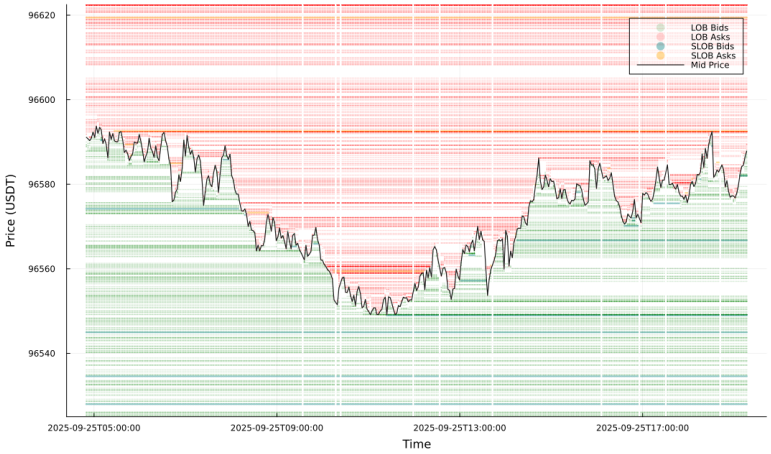


Figure 5: Spatiotemporal liquidity distribution showing active LOB layers and latent SLOB.

Support, Resistance, and Price Discovery: The LOB forms dense, horizontal striations of resting bids and asks that act as structural support and resistance levels. The mid-price trajectory adheres strictly to continuous double auction mechanics; price movements are entirely endogenous consequences of liquidity consumption (erasing resting layers) rather than a stochastic overlay. **The Latent "Safety Gap":** Stop-loss orders are intentionally sparse and positioned at the outer boundaries of the dense LOB. This confirms the behavioral logic of strategic agents, who maintain a distinct buffer between the active trading zone and their risk-management triggers. **Liquidity Gaps and Fragility:** Localised liquidity voids arise naturally and coincide with rapid price changes. Because the price is anchored by dense liquidity bands, flash crashes cannot occur via a simple random walk. Instead, cascading failures require synchronised, pro-cyclical shocks—such as clustered stop-loss triggers—capable of instantly depleting multiple LOB layers.

Robustness Under Diverse Stop-Loss Regimes We evaluate CHAD under a broad stress-testing design involving heterogeneous agent ecologies, multiple look-back horizons $L \in \{50, 250, 500\}$, and varying stop-loss adoption rates $\rho_{SL} \in \{0.0, 0.01, 0.02, 0.05, 0.1, 0.2, 0.25, 0.3, 0.5, 1.0\}$. This setup produces a diverse set of market environments, ranging from low-friction liquidity conditions to highly reactive, stop-loss-dense regimes. Six representative ecologies are included, spanning zero-intelligence configurations $(3500, 0, 0, 0)$ through to hyper-algorithmic regimes $(2000, 1000, 500, 1)$, alongside mixed and liquidity-crisis structures, ensuring coverage across passive, transitional, and strategy-intensive market compositions.

Across all configurations, we observe consistent and bounded behaviour in standard stylised fact measures, including volatility, kurtosis, and return dynamics. While these metrics vary systematically with both ecological composition and stop-loss prevalence, they remain within empirically plausible ranges and do not exhibit explosive or pathological behaviour. Critically, no configuration leads to runaway dynamics, absorbing states, or structural collapse of the limit order book. Even under extreme stop-loss saturation, the system remains stable, with liquidity depletion events remaining endogenous and self-correcting rather than degenerative. To summarise the global behaviour of the system under this stress test, Table 10 reports aggregated ranges across all configurations.

These results suggest that while stop-loss mechanisms introduce structured variation in market dynamics, they do not destabilise the system. Instead, they modulate the distribution of returns within a bounded and interpretable range. This bounded variability suggests that the simulator maintains strong structural robustness under heterogeneous and adversarial conditions. In particular, the interaction between agent heterogeneity and latent stop-loss liquidity produces variation in market dynamics without compromising stability or realism.

Importantly, stop-loss mechanisms act as endogenous features of the market microstructure rather than external shocks. Their interaction with the limit order book generates additional variability and localised liquidity clustering without destabilising the underlying system, modulating

Table 10: Aggregate robustness summary across all ecologies, stop-loss regimes, and look-back horizons (3,600 simulations).

Metric	**Min**	**Max**	**Interpretation**
Volatility (relative units)	0.92	1.41	Bounded variation across regimes
Kurtosis	15.2	27.8	Heavy tails without unwieldy divergence
Mean Total Error $J(\theta)$	11.7	16.0	Stable compliance with empirical error
Validity Score (%)	64	90	Gradual degradation under saturation

the distribution of returns in a controlled and economically interpretable manner.

Within the context of agent-based financial modelling, these results demonstrate that the simulator generates systematic variation in higher-order moments endogenously, without relying on heavy-tailed exogenous shock distributions. Instead, features like extreme kurtosis emerge organically from the interplay between order book mechanics and heterogeneous strategies.

Overall, under a wide and intentionally stress-tested parameter space, CHAD exhibits no evidence of pathological divergence. The model produces realistic, heterogeneous market behaviour while maintaining numerical and structural stability across all evaluated regimes. This provides strong support for its use as a calibrated simulator for downstream experiments involving endogenous risk mechanisms and market microstructure interventions.

7 Conclusion

In this paper we introduced CHAD (Complex Heterogeneous Agent Dynamics), a high-performance agent-based financial market simulator designed to reproduce realistic market microstructure through fully endogenous agent interaction. By combining a heterogeneous agent ecology with a deterministic pre-scheduled execution architecture, CHAD generates market dynamics without reliance on exogenous price processes, historical replay, or fundamental-value guidance.

Three primary contributions emerge from this work. First, we demonstrate that a fully synthetic market ecology can reproduce a broad range of empirical stylised facts, including heavy-tailed returns, volatility clustering, realistic spread dynamics, and liquidity-driven price formation. These properties arise from the interaction of heterogeneous agents operating through a continuous double-auction mechanism rather than from externally imposed stochastic processes.

Second, we introduce a dual-book market structure consisting of a public Limit Order Book (LOB) and a private Stop-Loss Order Book (SLOB). This framework formalises latent liquidity as a first-class market component, enabling explicit modelling of hidden risk concentration, recursive liquidation cascades, and stop-loss-driven fragility. To our knowledge, this represents the first agent-based market simulator to incorporate a dedicated stop-loss liquidity layer while maintaining fully endogenous price formation.

Finally, we note that CHAD introduces a pre-scheduled execution architecture that removes the dynamic event-queue bottlenecks common in traditional agent-based simulators. Computational benchmarks demonstrate near-linear scaling to one million agents, enabling large-scale ecological sweeps and sensitivity analyses that are difficult to perform in existing event-driven frameworks.

CHAD, therefore, provides both a scalable simulation engine and a novel microstructural framework for studying systemic risk, hidden liquidity, and emergent market behaviour. More broadly, CHAD shows the ability to simulate many higher-order market phenomena that were traditionally initialised by exogenous shocks when studied in former ABMs, instead these emerge naturally from the interaction of heterogeneous agents and latent liquidity within our simulator when a sufficiently rich market microstructure is present.

References

[1] Amrouni, S., Moulin, A., Balch, T.: CTMSTOU driven markets: simulated environment for regime-awareness in trading policies. arXiv preprint arXiv:2202.00941 (2022)

[2] Assefa, S.A., Dervovic, D., Mahfouz, M., Tillman, R.E., Reddy, P., Veloso, M.: Generating synthetic data in finance: opportunities, challenges and pitfalls. In: Proceedings of the first ACM international conference on AI in finance. pp. 1–8 (2020)

[3] Avellaneda, M., Stoikov, S.: High-frequency trading in a limit order book. Quantitative Finance **8**(3), 217–224 (2008)

[4] Belcak, P., Calliess, J.P., Zohren, S.: 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. pp. 42–56. Springer (2021)

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

[6] Christie, A.A.: The stochastic behavior of common stock variances: Value, leverage and interest rate effects. Journal of financial Economics **10**(4), 407–432 (1982)

[7] Cliff, D.: An open-source limit-order-book exchange for teaching and research. In: 2018 IEEE Symposium Series on Computational Intelligence (SSCI). pp. 1853–1860. IEEE (2018)

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

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

[10] 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)

[11] 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)

[12] Gould, M.D., Porter, M.A., Williams, S., McDonald, M., Fenn, D.J., Howison, S.D.: Limit order books. Quantitative Finance **13**(11), 1709–1742 (2013)

[13] Kobayashi, S., Hashimoto, T.: Benefits and limits of circuit breaker: Institutional design using artificial futures market. Evolutionary and Institutional Economics Review **7**, 355–372 (2011)

[14] LeBaron, B., Arthur, W.B., Palmer, R.: Time series properties of an artificial stock market. Journal of Economic Dynamics and control **23**(9-10), 1487–1516 (1999)

[15] Liston, P., Gretton, C., Lensky, A.: The role of stop-loss orders in market efficiency and stability: An agent-based study. In: Proceedings of the 16th International Conference on Agents and Artificial Intelligence - Volume 1: ICAART. pp. 280–288. SciTePress (2024). https://doi.org/10.5220/0012371400003636

[16] Mizuta, T.: A brief review of recent artificial market simulation (agent-based model) studies for financial market regulations and/or rules. Available at SSRN 2710495 (2016)

[17] Mizuta, T., Hayakawa, S., Izumi, K., Yoshimura, S.: Investigation of relationship between tick size and trading volume of markets using artificial market simulations. JPX working paper (2) (2013)

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

[19] Palmer, R.G., Arthur, W.B., Holland, J.H., LeBaron, B., Tayler, P.: Artificial economic life: a simple model of a stockmarket. Physica D: Nonlinear Phenomena **75**(1-3), 264–274 (1994)

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

[21] Todd, A., Beling, P., Scherer, W., Yang, S.Y.: Agent-based financial markets: A review of the methodology and domain. In: 2016 IEEE Symposium Series on Computational Intelligence (SSCI). pp. 1–5 (2016). https://doi.org/10.1109/SSCI.2016.7850016

[22] Winker, P., Gilli, M., Jeleskovic, V.: An objective function for simulation based inference on exchange rate data. Journal of Economic Interaction and Coordination **2**(2), 125–145 (2007)

[23] Xiong, X., Liu, J., Yang, Z., Han, J., et al.: An agent-based model for the impact of price limit changes on market quality. International Journal of Information Technology & Decision Making (IJITDM) **21**(06), 1777–1795 (2022)