SpikeTrade: Simulating Stock Market Behavior with Spiking Neural Networks

Soham Gupta
Computer Science and Statistics
Harvard University
sohamgupta@college.harvard.edu

Abstract

Financial markets are shaped not only by algorithms and data but also by the nuanced, emotion-driven decisions of human traders. This project investigates whether Spiking Neural Networks (SNNs)—a biologically inspired neural model—can be used to simulate human-like financial decision-making at scale. We build a hybrid market simulation composed of SNN-driven agents, algorithmic traders leveraging technical indicators, and random traders mimicking noise-driven liquidity providers. Results show that in sectors such as Finance and Consumer, the simulation closely replicates historical market behavior, achieving R-squared values above 70% in several stocks. Even where R-squared values were lower, many simulated price trends qualitatively mirrored real-world movements, suggesting that the framework successfully captures important behavioral and structural features of the market. An interactive web dashboard further demonstrates the simulation's capabilities, enabling real-time experimentation. These findings support the hypothesis that SNNs can model key aspects of retail investor behavior and lay the foundation for more human-aware market simulations.

1 Introduction

Financial decision-making is a complex cognitive process, involving dynamic assessments of risk, reward, and sentiment. While traditional market simulations often focus on purely algorithmic or statistically driven agents, real-world trading behavior—particularly among retail investors—is influenced by emotion, prior experience, and heuristic thinking. Neuroscientific research has shown that decision-making under risk activates brain regions, supporting the idea that biological processes play a central role in economic behavior.

Spiking Neural Networks (SNNs), which emulate neuron-level activity through spike-based communication and plasticity, offer a promising framework for modeling such behavior. Unlike traditional neural networks, SNNs encode information temporally and react to asynchronous events—properties that align well with the noisy, time-sensitive nature of financial markets. Work has demonstrated the use of SNNs in simulating decision-making circuits in the brain and in modeling financial time series. However, their application to large-scale simulations of investor behavior remains underexplored.

In this paper, we pose the following question: How can Spiking Neural Networks be used to model human decision-making in financial markets, capturing the temporally dependent and emotionally driven trading behaviors of retail investors through large-scale simulations? We hypothesize that an SNN-driven approach will exhibit sufficient fidelity to replicate at least 70% of historical stock price movements for a U.S.-based equity heavily influenced by retail traders.

To test this hypothesis, I developed a comprehensive market simulation composed of three trader types—SNN-based agents, rule-based algorithmic traders using technical indicators like RSI and

Preprint. Under review.

MACD, and stochastic random traders that introduce liquidity and noise. We simulate market dynamics across sectors and volatility regimes, using Amihud's illiquidity metric to realistically model the impact of aggregate trader activity on price. Performance is evaluated through both quantitative metrics (R-squared, RMSE, volume correlation) and qualitative alignment with historical market patterns. Our findings suggest that even when statistical metrics fall short, the simulation can still produce structurally valid price dynamics, reinforcing the viability of biologically inspired financial modeling.

2 Related Works

The validity of our approach integrates insights from neuroscience, machine learning, and behavioral finance. Spiking Neural Networks (SNNs) offer a biologically grounded framework for modeling human trading behavior by mimicking spike-based neural processes, making them well-suited for tasks that involve temporal dependencies, adaptive decision-making, and emotional variability.

2.1 Neuroscience of Financial Decision-Making

Neuroscience has provided critical insights into how humans make financial decisions. fMRI studies reveal that professional traders exhibit asymmetric activation in the prefrontal cortex (PFC) during strategic planning, while suppressing amygdala activity during loss anticipation [1]. The nucleus accumbens (NAcc), a brain region involved in reward processing, activates prior to risk-seeking decisions, further reinforcing the idea that emotional and reward-based processes shape investment behavior [2].

Additionally, work has linked anterior cingulate cortex (ACC) activity to error detection and learning in risk-based tasks, reinforcing its importance in adjusting trading strategies based on feedback [2]. Furthermore, the somatic marker hypothesis proposes that bodily signals guide decision-making in uncertain environments such as markets, highlighting the physiological underpinnings of economic behavior [10].

2.2 Machine Learning Advances in SNNs

Traditional machine learning models, such as LSTMs and RNNs, are limited in their ability to model event-based, irregular financial data. SNNs offer a more biologically plausible solution by using event-driven computation and encoding temporal information in spike trains. Surrogate gradient learning has been a key breakthrough, enabling efficient backpropagation in spiking networks [3].

SNNs have also been used to simulate brain structures involved in decision-making. For instance, a biologically constrained SNN model of the primate basal ganglia accurately reproduced observed decision patterns and action selection behavior [4], supporting the hypothesis that SNNs can mimic human-like decision systems.

2.3 Applying Neuroscience and SNNs to Finance

The intersection of neuroscience and finance has given rise to the field of neurofinance, which explores how cognitive biases, emotional responses, and neural mechanisms influence investor behavior. Frydman and Rangel, for example, demonstrated that reducing the saliency of realized gains and losses could mitigate the disposition effect—the tendency for investors to sell winning stocks too early and hold onto losing ones [5]. This aligns closely with the activity of reward-related neural systems such as the nucleus accumbens (NAcc), supporting the notion that emotional reinforcement plays a central role in financial decision-making.

Expanding on this, Srivastava et al. reviewed how neurobiological and psychological factors affect financial behavior, concluding that biologically informed models can offer more realistic simulations of investor decision processes [6].

Recently, SNNs have shown significant promise in financial applications. AbouHassan et al. trained SNNs on multimodal data streams, including financial time series and news sentiment, to produce predictive and interpretable models of market behavior [7]. Their study highlighted not only strong predictive performance but also transparency in decision-making by examining spike-based input activation.

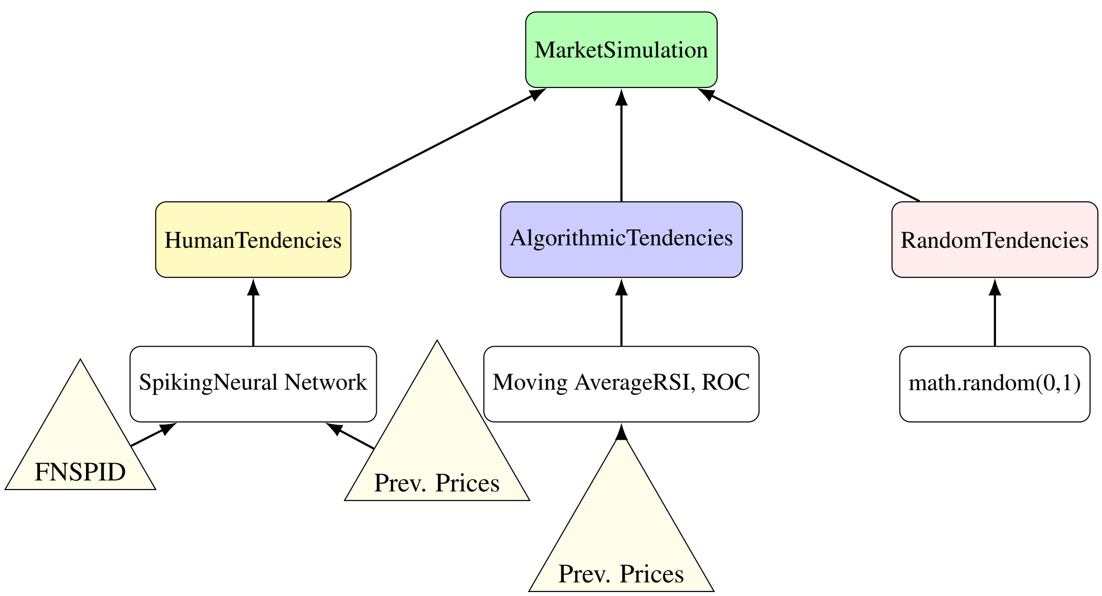
Complementary to this, Reid et al. showed that SNNs can outperform traditional neural networks on tasks involving temporal financial data, reinforcing their suitability for time-sensitive prediction problems [9].

Finally, integrating with cutting-edge neuromorphic hardware, Yakopcic et al. demonstrated the feasibility of deploying SNNs on Intel's Loihi chip to perform high-speed cognitive asset allocation tasks, offering scalable and energy-efficient solutions for finance [8]. Khan further emphasized that SNNs are well-positioned for the future of brain-inspired computing in finance due to their unique combination of biological fidelity, explainability, and hardware compatibility [11].

Together, these developments suggest that SNNs are not only biologically aligned with the mechanisms underlying financial decision-making but also technically viable for deployment in large-scale financial systems.

By integrating neuroscience insights and the computational structure of SNNs, we can simulate human traders who react to news, price trends, and behavioral biases. These traders, when placed in a larger market simulation, offer a powerful tool to study how human-like decision-making impacts aggregate market behavior.

3 Methodology



3.1 Project Overview

In the stock market, there are three primary types of participants: human investors, algorithmic trading bots, and market makers.

Human investors make decisions based on a combination of stock analysis, price movements, financial news, and, importantly, emotions. Their strategies vary in time horizon and their capital ranges from individual savings to large institutional funds. Human behavior in markets is often influenced by sentiment, cognitive biases, and reactions to external events.

Algorithmic trading bots are automated systems designed to execute trades based on predefined strategies, typically driven by patterns in price movements or technical indicators. These bots focus on exploiting short-term price fluctuations and generally ignore a stock's intrinsic value. They use complex algorithms to detect trends and execute trades at high speeds, capitalizing on small, rapid opportunities that are difficult for humans to catch.

Market makers provide liquidity to the market by constantly quoting buy and sell prices. They earn profits from small price differentials and ensure that all participants, including human investors and algorithmic bots, can execute trades without delay. Due to the scale and speed of their activity, their actions can appear unpredictable and are effectively random in the realm of simulation.

To create an accurate and scalable market simulation, I modeled all three trader types. Since the simulation operates at large scale, it is sufficient to represent each class of trader with an average behavioral model.

Human traders, the core focus of this project, were modeled using a Spiking Neural Network (SNN) trained on stock price data, earnings reports, and financial news. This design aims to replicate the decision-making patterns of retail investors who trade based on both rational information and emotional responses.

Algorithmic traders were implemented using simple yet widely adopted strategies such as Moving Average Crossovers and Rate of Change (ROC) indicators. These heuristics, while not optimal, are common in real-world trading and produce credible approximations of algorithmic behavior.

Market makers were simulated using a randomized trading strategy to reflect their high-frequency, near-random movement.

3.2 Implementation

3.2.1 Data Pipeline

To support the training and evaluation of the Spiking Neural Network (SNN) and Algorithmic trading bot, I developed a dedicated data loading and preprocessing module—`FinancialDataLoader`—which orchestrates the integration of market data, financial news sentiment, and earnings announcements into a unified input format compatible with spike-based modeling.

Dataset Sources. The pipeline incorporates data from the following key sources:

- **Yahoo Finance (via `yfinance`):** Accessed from Python library and used to retrieve company's market fundamentals for a given stock symbol.
- **FNSPID Dataset:** This includes historical stock prices and sentiment-scored financial news articles for NASDAQ-listed companies from 2023–2024. Data was accessed and downloaded from HuggingFace-hosted endpoints.

News Data Processing. News articles are loaded from a CSV file from FNSPID and filtered by stock symbol and date range. Sentiment scores are resampled to a daily frequency and aligned with trading days. This creates a continuous signal that reflects public sentiment for each day, forming the input to the `news_input` layer of the SNN.

Earnings Signal Encoding. Earnings data is fetched using the `yfinance` python library. For each earnings report, a binary indicator is generated marking the earnings announcement day and the five subsequent trading days. This event-driven encoding is passed to the `earnings_input` layer to simulate the market's reaction window following key financial disclosures.

Market Feature Engineering. Market data includes commonly used technical indicators such as:

- Daily returns, Simple Moving Averages (SMA-20, SMA-50)
- Relative Strength Index (RSI), MACD and signal line
- Bollinger Bands, trading volume, and Average True Range (ATR)

Combining Inputs. The `combine_market_data` method synchronizes the three input streams—market indicators, sentiment signals, and earnings events—into a single `DataFrame` indexed by trading day.

3.3 Spiking Neural Network Training

The core component of this project is a custom-built Spiking Neural Network (SNN) designed to simulate human-like trading decisions based on temporally structured financial inputs. The network was implemented using the `BindsNET` framework and is defined in the `TradingSNN` class in `neural_network.py`, extending BindsNET's base `Network` class.

Spike Conversion. Prior to feeding the data into the SNN, the continuous inputs are normalized and encoded into spike patterns. While this conversion occurs downstream during training, the unified data structure established by the loader ensures the correct temporal alignment of all signals.

This modular pipeline enables efficient, scalable preprocessing and lays the foundation for interpretable, biologically grounded trading simulations. The integration of multiple financial signals mimics the information sources that real human investors rely on, thereby improving the realism and behavioral fidelity of the SNN's decision-making process.

Architecture Overview. The network processes three distinct streams of financial data—market indicators, news sentiment, and earnings events—each encoded into separate input pathways:

- **Market Input Layer (10 dimensions):** Includes daily return, short- and long-term simple moving averages (SMA-20 and SMA-50), Relative Strength Index (RSI), MACD and signal line, Bollinger Bands (upper and lower), trading volume, and Average True Range (ATR).
- **News Input Layer (1 dimension):** Captures sentiment scores derived from the FNSPID dataset, normalized and scaled to reflect external market influence.
- **Earnings Input Layer (1 dimension):** Encodes binary indicators for earnings announcements, highlighting key event windows that often trigger retail investor reactions.

Neural Structure. The internal network comprises:

- A hidden layer of 100 Leaky Integrate-and-Fire (LIF) neurons, configured with trace-enabled synaptic memory for STDP learning.
- An output layer of two LIF neurons, representing binary trading decisions: `buy` and `sell`.

Learning Dynamics. All synaptic connections—from inputs to hidden, and from hidden to output—use the `PostPre` STDP learning rule with a learning rate of 1×10^{-4} and no weight decay. This biologically inspired mechanism allows the network to adapt its weights based on the relative timing of spikes in pre- and post-synaptic neurons.

Information Processing. At each timestep, the network receives three synchronized spike-encoded tensors corresponding to the market, news, and earnings data. These are processed via the `forward` method, which updates internal voltages and triggers spikes across the layers. Decision-making is based on the relative spike count of the output neurons: if the `buy` neuron spikes more than `sell`, the model outputs a buy signal; if the reverse is true, it signals a sell; otherwise, it returns a hold decision.

3.4 Algorithmic Trader

To complement the SNN trader, I developed a rule-based algorithmic trading system grounded in widely used technical indicators. This system executes trades based on predefined market signals derived from historical stock price data accessed via the `yfinance` API.

Data and Indicators. The algorithm relies on standard OHLCV (Open, High, Low, Close, Volume) price data and applies a suite of technical indicators commonly used in quantitative trading:

- **Moving Average Convergence Divergence (MACD)**: Measures trend direction and momentum based on exponential moving averages. Includes MACD line, signal line, and histogram.
- **Relative Strength Index (RSI)**: Detects overbought or oversold conditions by analyzing recent price gains and losses.
- **Bollinger Bands**: Defines dynamic support and resistance levels using rolling standard deviations around a moving average.
- **Average True Range (ATR)**: Captures volatility by averaging the daily trading range over a fixed window.
- **Volatility Metrics**: Includes daily and annualized volatility, rolling standard deviation, maximum drawdown, and Value at Risk (VaR) at 95% and 99% confidence intervals.

Market Regime Detection. A dedicated function classifies the current market state into one of three regimes: `trending`, `ranging`, or `volatile`. This classification is based on rolling trend strength and volatility calculations, allowing for regime-aware strategy adjustments.

Signal Generation. The trading logic combines signals from the above indicators to identify buy and sell opportunities. For example:

- A crossover of the MACD line above the signal line may indicate a buy.
- RSI values above 70 or below 30 flag potential reversal zones.
- Price interactions with Bollinger Bands can imply breakouts or reversion to the mean.

This implementation of the algorithmic trader offers an interpretable foundation for algorithmic trading that reflects techniques used by retail and institutional traders alike. While relatively simple compared to machine learning-based models, it provides a useful baseline and approximation for algorithm based trading bots.

Random Trader. To model market participants such as high-frequency market makers or noise traders whose actions are not easily predicted or driven by fundamental analysis, I implemented a `RandomTrader` agent. This trader selects its actions—`buy`, `sell`, or `hold`—randomly at each timestep, with configurable probabilities (e.g., 30% buy, 30% sell, 40% hold). While simplistic, this agent plays an important role in the simulation by introducing stochastic liquidity and mimicking the unpredictable behavior of non-strategic actors in financial markets.

3.5 Simulation

The final stage of this project integrates all trader types—Spiking Neural Networks, algorithmic bots, and random traders—into a unified market simulation framework. I elected to do a stratified simulation by grouping stocks into sectors or themes and simulating their dynamics in parallel. Each simulated stock is driven by a shared set of traders, creating realistic co-movement and correlation patterns.

Simulation Engine. The simulation runs over a configurable number of days (e.g., 100) and generates synthetic time series for both `price` and `volume`. The engine begins from the most recent date in the historical dataset and uses historical volatility as a baseline. Daily returns are sampled from a Gaussian distribution, and volumes are perturbed with random noise to reflect natural variability.

Stock Grouping and Correlation. Stocks are organized into groups (e.g., tech, healthcare), and each group is simulated independently. After simulation, average pairwise return correlations are computed to evaluate whether stocks within a group move together, as they often do in real markets.

Price and Volume Generation. For each stock, the simulation tracks:

- **Prices:** Initialized from real historical prices and updated using daily percentage returns sampled from the historical return distribution.
- **Volumes:** Modeled as multiplicative shocks around prior-day volume, preserving noise and trend characteristics.

Trader Integration. Each trader operates on the most recent market data and generates a trading decision—`buy`, `sell`, or `hold`—which influences both price and volume evolution. To more realistically simulate how trading activity affects asset prices, I incorporated Amihud's illiquidity metric to quantify market impact. For each stock, the illiquidity measure is computed as the average ratio of absolute return to dollar volume, scaled appropriately. During each simulated day, the total net order flow (difference between aggregate buy and sell volumes across all traders) is used to compute a market impact factor. This impact is proportional to trade size, inversely proportional to market liquidity, and scaled by price, following the formula:

$$\text{Impact} = \lambda \cdot \left(\frac{Q}{V}\right) \cdot P$$

where λ is the illiquidity measure, Q is the net trade volume, V is the daily volume, and P is the price. The resulting impact adjusts the price upward or downward depending on net order direction, allowing the model to reflect the temporary influence of large trades in illiquid markets. This allows me to model the price changes based on trading volume at the day level. This works well for my use case because I am simulating over the time span of days which fits the Amihud metric.

Performance Evaluation. After simulation, results are analyzed using:

- **Correlation** between simulated and historical returns
- **Root Mean Squared Error (RMSE)** of return paths
- **Volume correlation** across time

Visualization. The system also supports plotting actual vs. simulated time series for both price and volume, enabling visual comparison across stock groups. These diagnostic plots highlight how well the simulation captures real-world trading patterns and market behaviors and reveal patterns that technical formulas often miss.

3.6 Interactive Simulation Dashboard

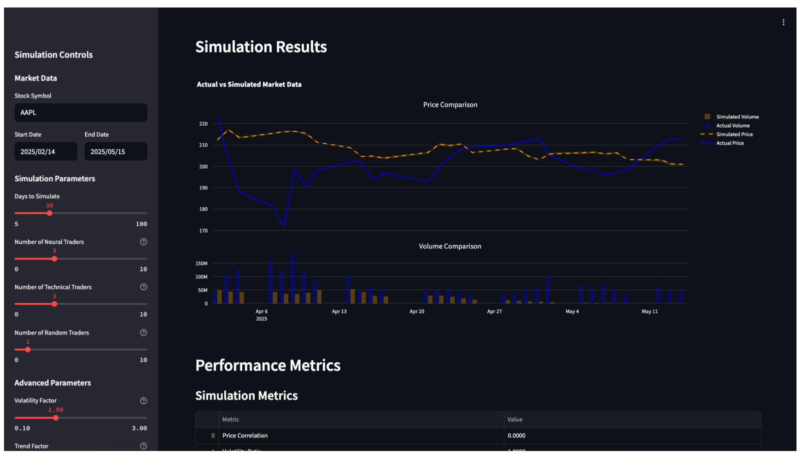


Figure 1: Interactive Web Simulation with adjustable parameters

To make the system more interactive and demonstrate the capabilities of the simulation engine, I built a web-based dashboard (Figure 2) that allows users to experiment with market simulations in real time. Built with streamlit, the interface lets users define the simulation window and adjust trader composition across neural, technical, and random agents. The simulation results are rendered immediately on a price-volume chart in comparison to the actual pricing results. By allowing real-time parameter adjustments and visual comparisons with historical data, I built the dashboard to enhance the interpretability and accessibility of the model.

4 Results

To evaluate the performance and realism of the market simulation framework, I conducted experiments across multiple stock groups categorized by sector (e.g., Consumer, Energy, Finance, Healthcare, Technology) and by market regime (e.g., High, Medium, Low Volatility). For each stock, both price and volume trajectories were simulated and visually compared against historical data over the same timeframe.

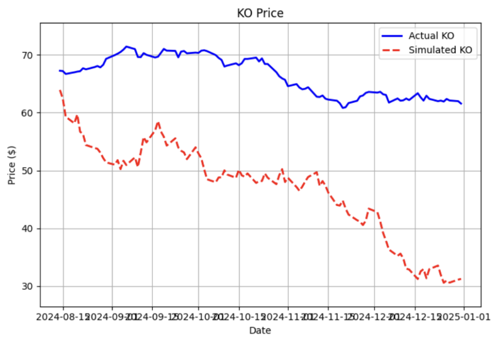


Figure 2: Simulated and Actual Price of KO Stock

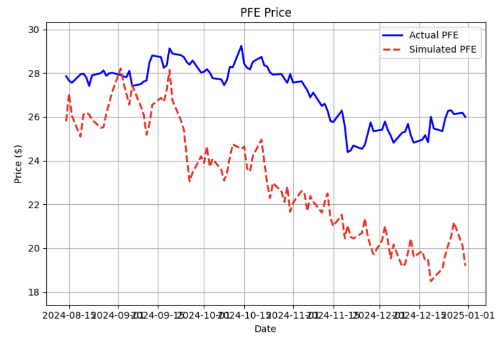


Figure 3: Simulated and Actual Price of PFE Stock

Price Trends. Across many sectors, simulated prices preserved general directional consistency (e.g., downward or sideways movements) with their real counterparts. In many cases, simulated trajectories followed macro trends but lacked the granularity of day-to-day volatility or abrupt shocks. For instance, in the Consumer and Healthcare groups, stocks such as KO and PFE mirrored their real-world trends with a reasonable degree of alignment (see Figures 3 and 4).

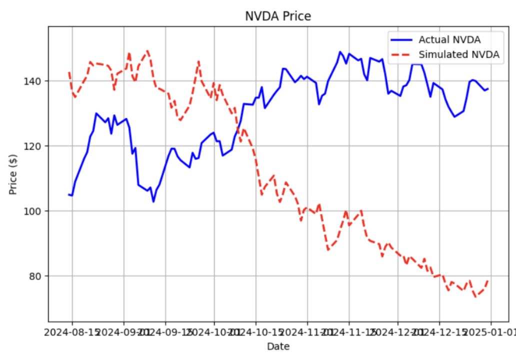


Figure 4: Simulated and Actual Price of NVDA Stock

However, there were also some sectors and stocks which showed adverse results, with the simulation price movement directly conflicting with the actual change in prices over time. NVDA (Figure 5) is a good example of this trend (which was interestingly very apparent in the technology sector).

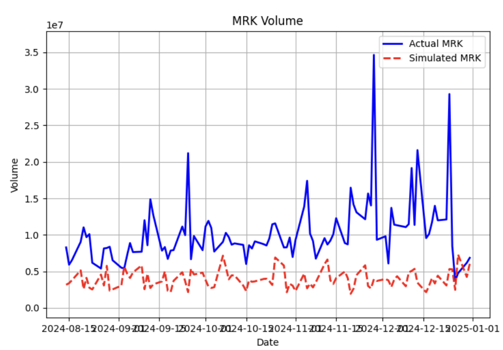


Figure 5: Simulation Volume for MRK

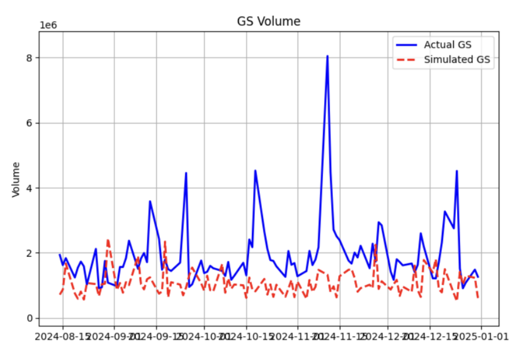


Figure 6: Simulation Volume for GS

Volume Dynamics. Simulated trading volumes were notably smoother and lower in magnitude compared to actual volume data, which displayed frequent spikes due to earnings reports, macro events, or institutional rebalancing (Figures 6, 7). The volume component did not yet incorporate a complex response to information flow limiting its realism.

Sector	Mean R^2	Best Performing Stock (R^2)	Worst Performing Stock (R^2)
Technology	32.7%	NVDA (61.5%)	INTC (10.8%)
Healthcare	36.0%	PFE (66.4%)	UNH (13.9%)
Finance	70.1%	WFD (83.5%)	BAC (57.6%)
Consumer	60.1%	WMT (86.3%)	PG (0.8%)
Energy	5.3%	SLB (13.4%)	EOG (0.1%)

Table 1: Quantitative Simulation Performance by Sector

Sector-Wise Observations.

- **Technology Sector:** The simulation achieved a mean R-squared of 32.7% across stocks. NVDA was the best-performing ticker, with 61.5% of variance explained by the simulation, while INTC underperformed at just 10.8%.
- **Healthcare Sector:** Simulated trends in healthcare were reasonably accurate, with a mean R-squared of 36.0%. PFE was among the best-fit stocks, with 66.4% of real price movement explained. In contrast, UNH performed poorly (13.9% explained), suggesting the model struggled with high-value, less volatile equities.
- **Finance Sector:** This was the best-performing sector overall, with a mean R-squared of 70.1%. WFC stood out with 83.5% explained variance. Even the weakest stock, BAC, maintained a relatively high R-squared of 57.6%.
- **Consumer Sector:** Simulation of consumer stocks yielded a mean R-squared of 60.1%. WMT was best matched (86.3% explained), but PG deviated substantially from real data, with only 0.8% of variance captured.
- **Energy Sector:** The simulation struggled most with energy stocks, achieving a mean R-squared of just 5.3%. SLB was the best (13.4% explained), while EOG had near-zero correlation (0.1%), indicating significant divergence in trend structure.

Volatility Regime Analysis. The simulation framework was also evaluated under high-, medium-, and low-volatility groupings. In high-volatility stocks like AMD and INTC, the simulation tended to underpredict variance and dampen short-term fluctuations. For low-volatility stocks such as ABBV or MRK, simulations better preserved real-world dynamics, likely due to fewer abrupt shifts and simpler return structures.

Quantitative Metrics. R-squared values provided an objective (but incomplete) benchmark for alignment between simulated and historical prices. While the Finance and Consumer sectors demonstrated strong correspondence, other sectors like Energy revealed the limitations of the current modeling approach. These differences suggest that the underlying trader logic and return generation process may need to be adapted per sector.

Please note that results graphs for all sectors and volatility groups are included in the Appendix below.

5 Discussion

5.1 Interpretation of Results

The simulation yielded encouraging results across multiple sectors and volatility regimes. In particular, the Finance and Consumer sectors demonstrated high predictive alignment, with R-squared values exceeding 80% for stocks like WMT and WFC. These results suggest that the agent-based simulation—composed of rule-based, spiking neural network, and random traders—can generate price movements that closely track real market behavior when applied to stable, liquid equities.

Importantly, even in cases where the R-squared values were modest (e.g., KO, PFE, Healthcare as a whole), the simulated price trajectories often captured realistic market dynamics. This included directional trends, inflection points, and volatility regimes that mirrored historical patterns, albeit with some deviation in timing or magnitude. These qualitative similarities are especially promising, as they indicate that the simulation framework is capable of reproducing the structural behaviors of markets even when the numerical fit is imperfect.

Such results highlight the dual value of the simulation: it can serve both as a quantitative predictor and as a qualitative modeling tool for understanding behavioral drivers of asset prices.

5.2 Limitations

Despite these strengths, several limitations became apparent in the evaluation:

- **Low Performance in Certain Sectors:** The Energy sector, in particular, exhibited very low R-squared values (e.g., 0.1% for EOG), revealing the model's difficulty in capturing stocks influenced by external macroeconomic or geopolitical factors not currently represented in the simulation.
- **Simplified Volume Modeling:** While price trajectories incorporated market impact via Amihud's illiquidity measure, volume was modeled with random perturbations and lacked feedback from trading activity, limiting realism in high-frequency environments.
- **Limited Adaptation by Traders:** Trader behavior was static throughout each simulation run, with no learning or adaptation mechanisms. This likely reduced the ability to capture shifting sentiment or structural breaks in time series.
- **One-Ticker Training Focus:** The spiking neural network was primarily trained on a small subset of stocks, reducing its generalizability across diverse tickers with different signal-to-noise characteristics.

5.3 Future Work

There are several promising directions to build upon the current simulation framework:

- **Sector-Specific Modeling:** Incorporate macroeconomic variables (e.g., oil prices, interest rates) and news sentiment specific to each sector to improve alignment in harder-to-model stocks like those in the Energy group.
- **Price-Volume Feedback Loop:** Enhance volume modeling by incorporating a feedback mechanism where trader activity affects both order flow and volatility, resulting in more realistic co-movement between price and volume.
- **Real-Time Dashboard Deployment:** Extend the existing simulation dashboard into a real-time experimental platform for testing trading hypotheses or demonstrating educational financial models interactively.

These improvements would help move the system from a proof-of-concept toward a more rigorous, scalable market simulation engine that integrates behavioral realism with quantitative performance.

References

[1] Ceravolo, M.G., Ragetti, G.M., Fattobene, L., and Dio, C.D. "Neural Correlates of Direct Access Trading in a Real Stock Market: An fMRI Investigation." *Frontiers in Neuroscience* 11 (2017): 536.

[2] Kuhnen, C.M., and Knutson, B. "The Neural Basis of Financial Risk Taking." *Neuron* (2005). https://doi.org/10.1016/j.neuron.2005.08.008.

[3] Neftci, E.O., Mostafa, H., and Zenke, F. "Surrogate Gradient Learning in Spiking Neural Networks: Bringing the Power of Gradient-Based Optimization to Spiking Neural Networks." *IEEE Signal Processing Magazine* 36.6 (2019): 51–63.

[4] Girard, B., Lienard, J., Gutierrez, C.E., Delord, B., and Doya, K. "A Biologically Constrained Spiking Neural Network Model of the Primate Basal Ganglia with Overlapping Pathways Exhibits Action Selection." *European Journal of Neuroscience* 58 (2021): 2413–2431.

[5] Frydman, C., and Rangel, A. "Debiasing the Disposition Effect by Reducing the Saliency of Information About a Stock's Purchase Price." *Journal of Economic Behavior & Organization* 107 (2014): 541–552.

[6] Srivastava, M., Sharma, G.D., and Srivastava, A.K. "Human Brain and Financial Behavior: A Neuroscience Perspective." *International Journal of Ethics and Systems* 2.1 (2019): 50–68.

[7] AbouHassan, I., Kasabov, N.K., Jagtap, V., and Kulkarni, P. "Spiking Neural Networks for Predictive and Explainable Modelling of Multimodal Streaming Data with a Case Study on Financial Time Series and Online News." *Scientific Reports* 13 (2023): Article number: 15125.

[8] Yakopcic, C., Rahman, N., Atahary, T., Taha, T.M., Beigh, A., and Douglass, S. "High Speed Cognitive Domain Ontologies for Asset Allocation Using Loihi Spiking Neurons." *arXiv preprint* arXiv:1906.12338 (2019).

[9] Reid, D., Hussain, A.J., and Tawfik, H. "Financial Time Series Prediction Using Spiking Neural Networks." *PLOS ONE* 9.8 (2014): e103656.

[10] Bechara, A., and Damasio, A.R. "The Somatic Marker Hypothesis: A Neural Theory of Economic Decision." *Games and Economic Behavior* 52.2 (2005): 336–372.

[11] Khan, A. "Spiking Neural Networks: The Future of Brain-Inspired Computing." *StatusNeo* (2025). Accessed May 13, 2025. `https://statusneo.com/spiking-neural-networks-the-future-of-brain-inspired-computing/`.

[12] Aroussi, Ran "yfinance: Yahoo! Finance market data downloader." *GitHub repository*, (2020). Available at: `https://github.com/ranaroussi/yfinance`

[13] Zihan, Dong, et al. "FNSPID: Financial News Sentiment and Price Impact Dataset for U.S. Stocks." *Hugging Face Datasets*, (2023). Available at: `https://huggingface.co/datasets/Zihan1004/FNSPID`

A Appendix

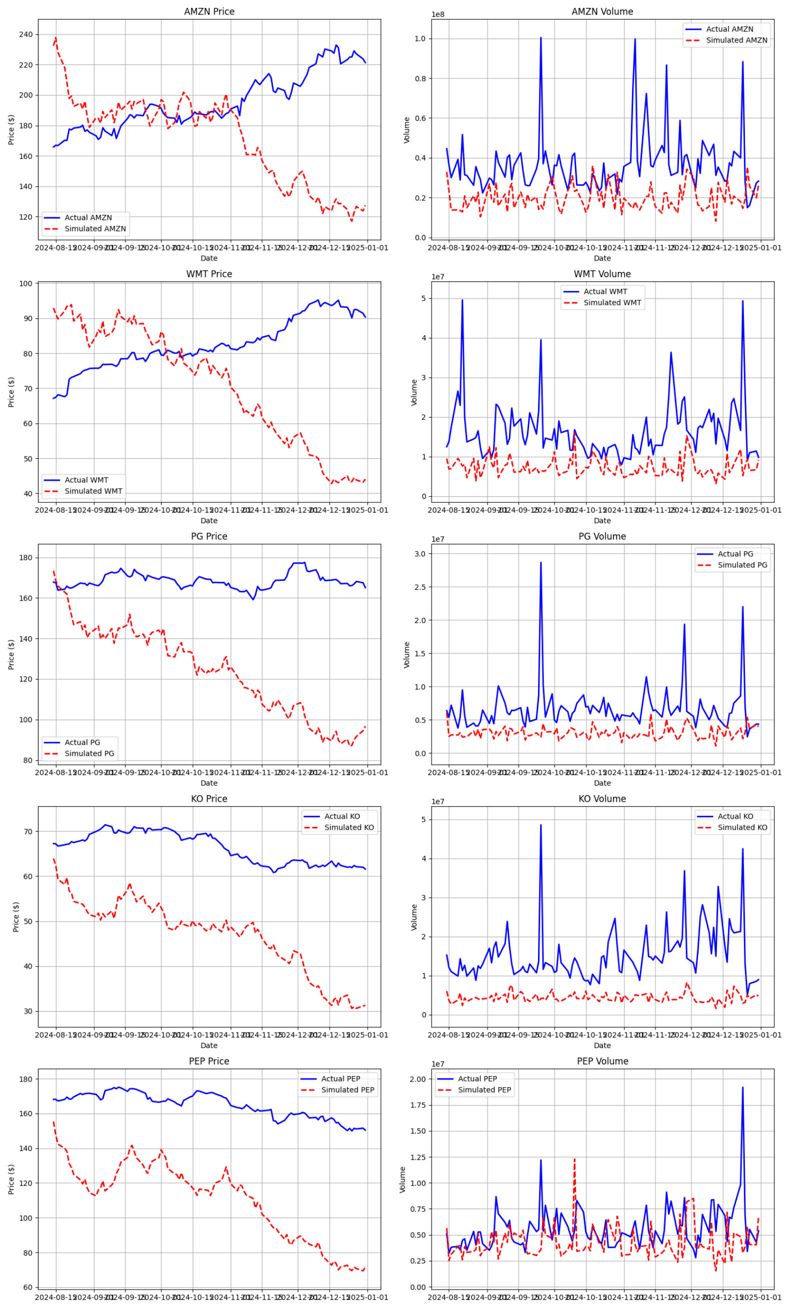


Figure 7: Consumer Sector Simulation Results

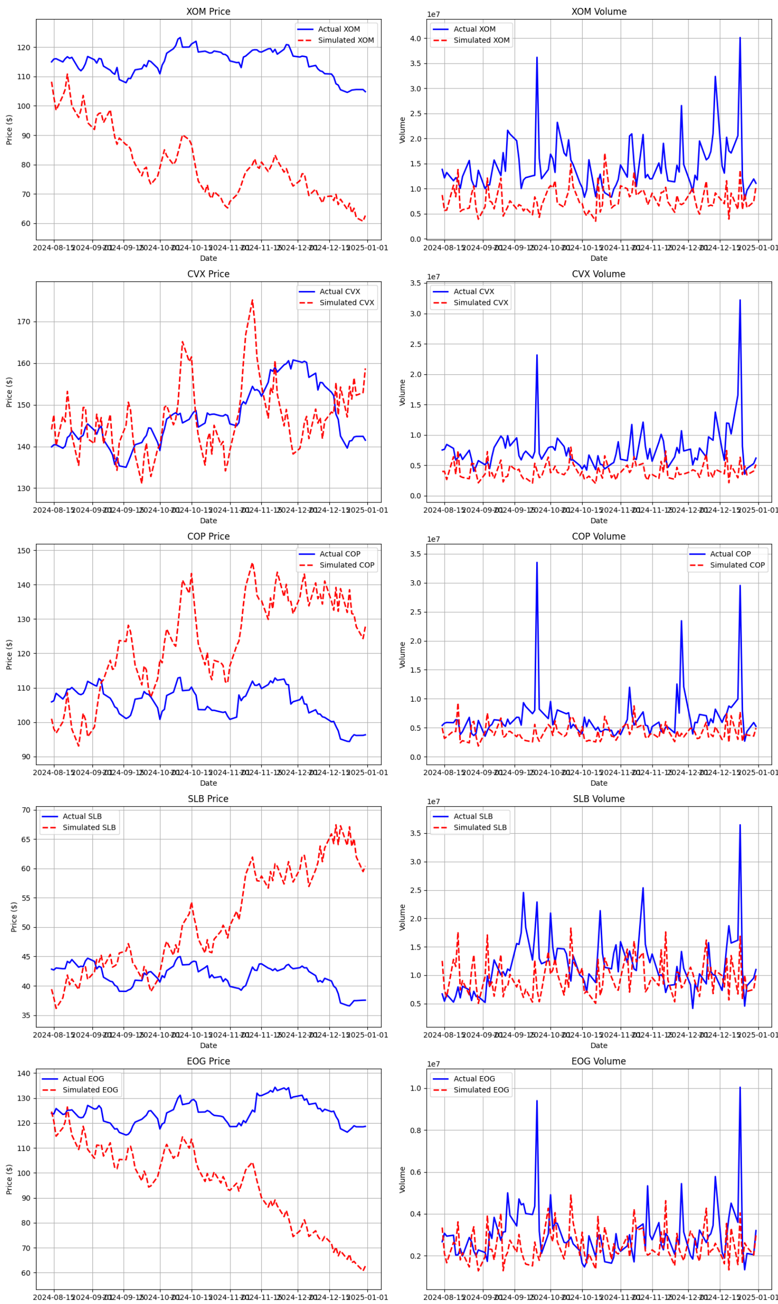


Figure 8: Energy Sector Simulation Results

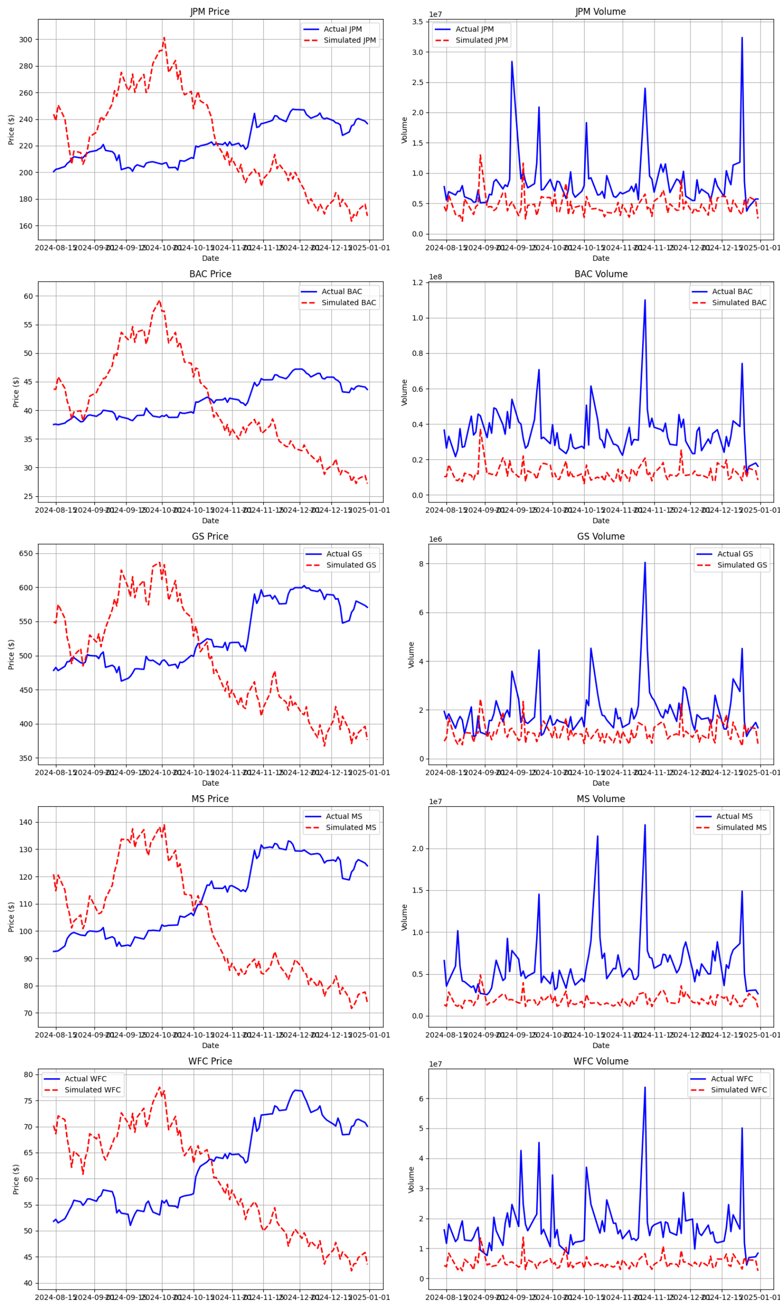


Figure 9: Finance Sector Simulation Results

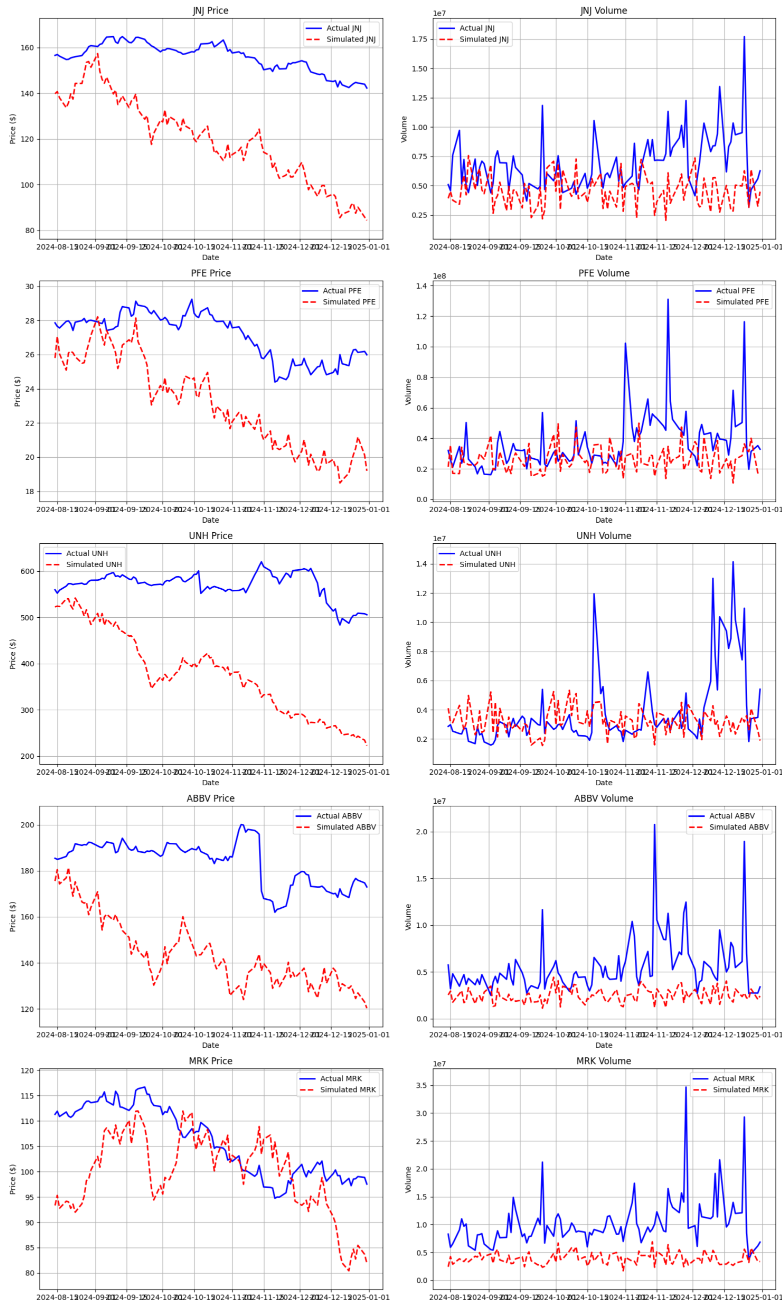


Figure 10: Healthcare Sector Simulation Results

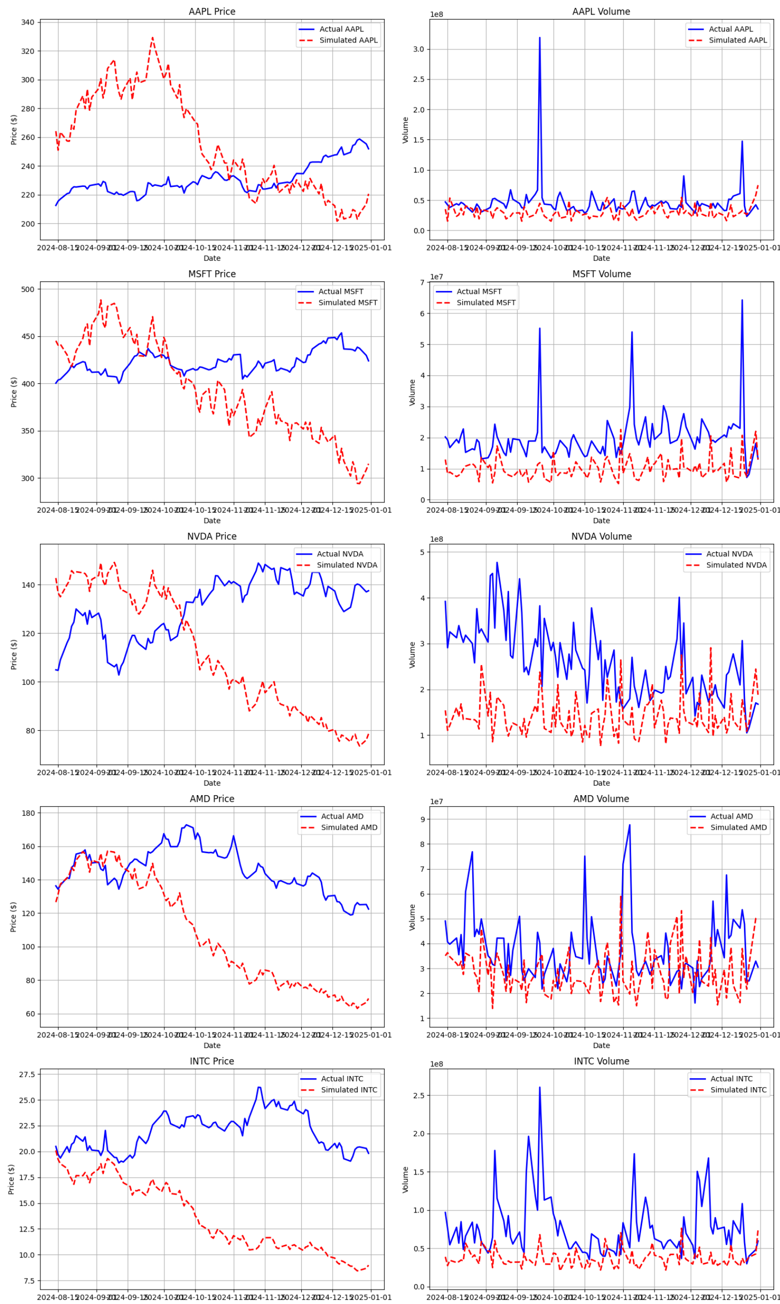


Figure 11: Technology Sector Simulation Results

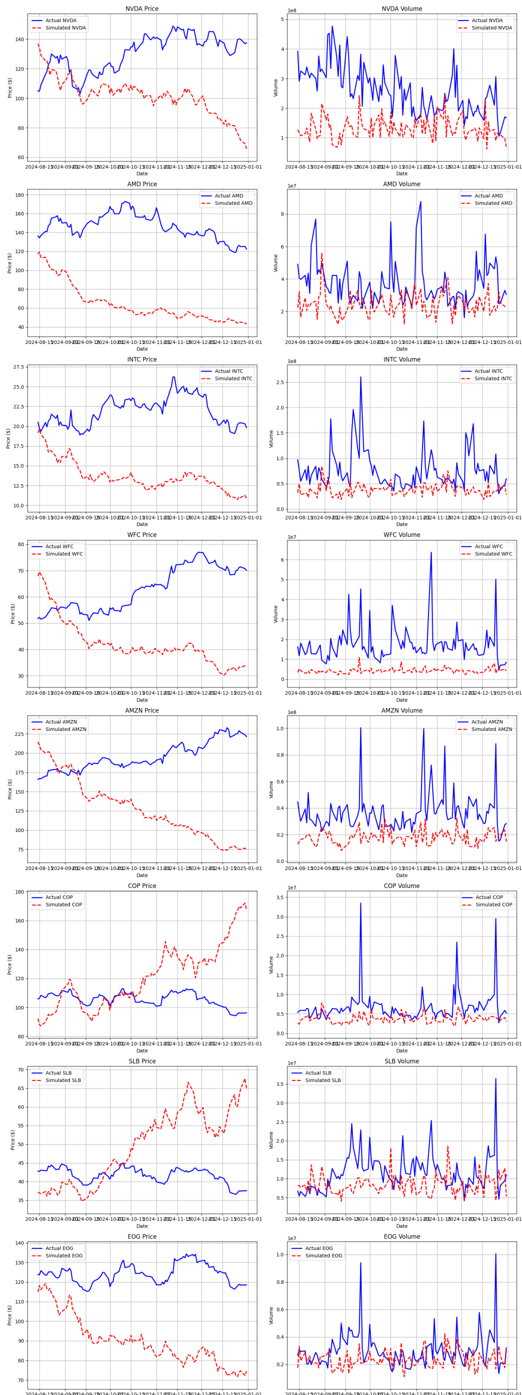


Figure 12: High Volatility Simulation Results

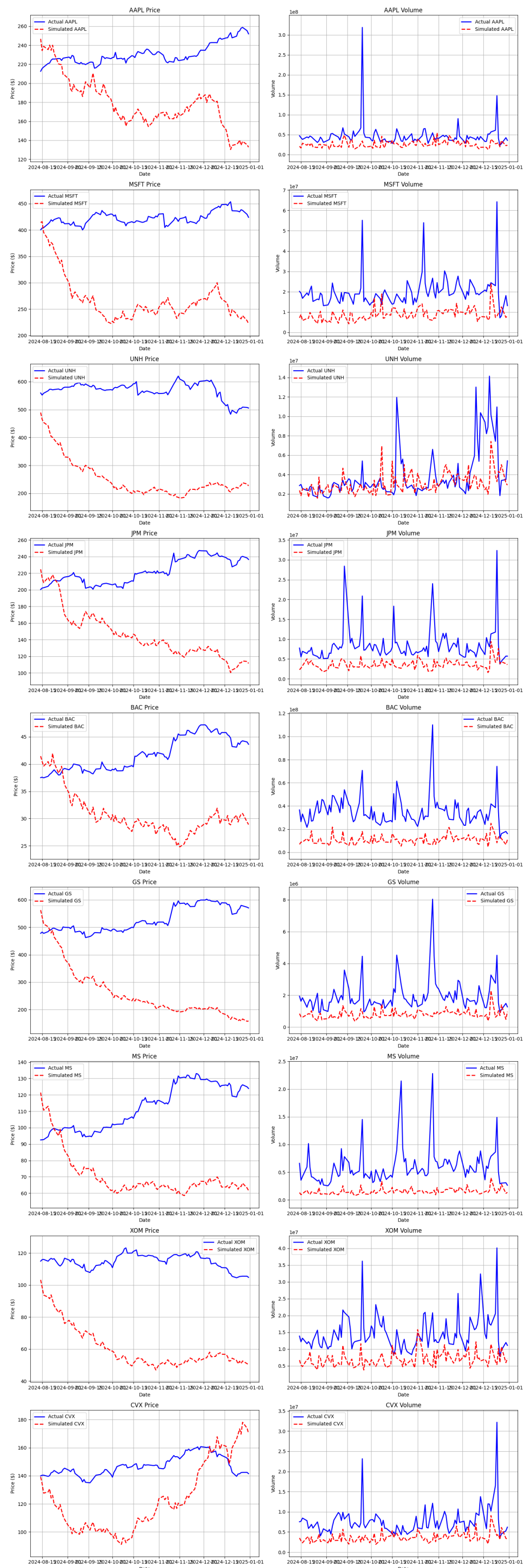


Figure 13: Medium Volatility Simulation Results

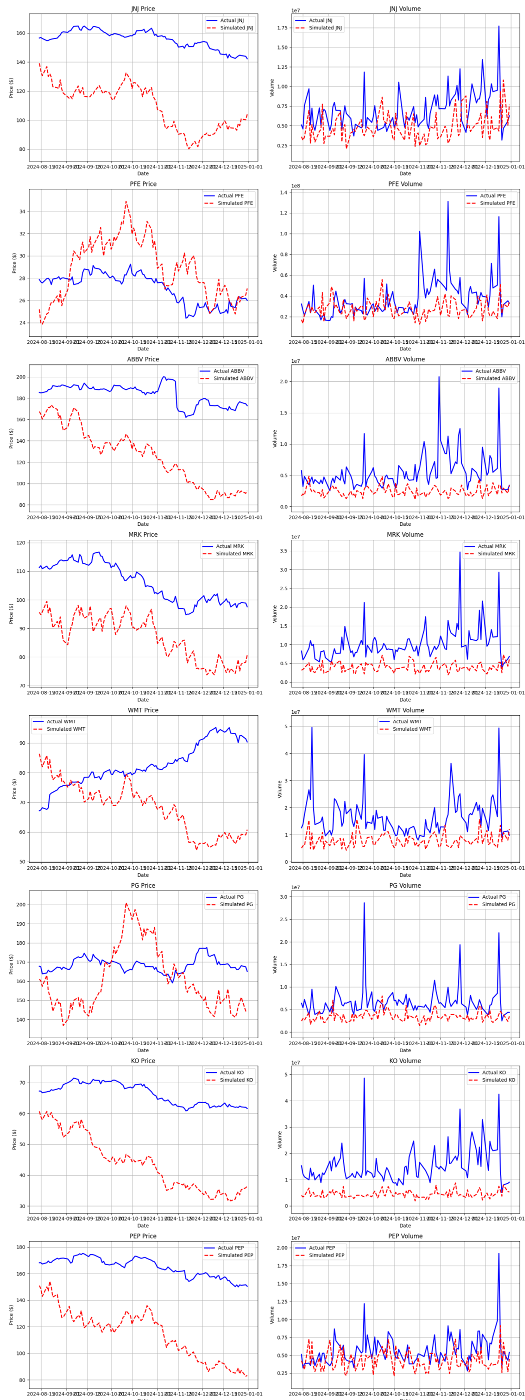


Figure 14: Low Volatility Simulation Results