Behavioral Reasoning Agent for Inference & Natural-language Risk Optimization in Trading

Zachary Pestrikov
California Institute of Technology
zpestrik@caltech.edu

Jeffrey Zhou
California Institute of Technology
jzhou3@caltech.edu

Ryan Hu
California Institute of Technology
rrhu@caltech.edu

Jacob Chang
California Institute of Technology
jchang8@caltech.edu

Abstract

Large Language Models (LLMs) have recently demonstrated an unprecedented ability to extract and reason over nuanced sentiment signals embedded in news, social media, and other unstructured text streams that move modern financial markets. Early domain-specific models such as FinBERT and BloombergGPT significantly improved sentiment classification accuracy but were conceived as advisory tools rather than autonomous decision-makers. More recent agentic frameworks such as FinMem and TradingAgents, leverage teams of LLMs that debate, recall layered memories, and coordinate trades. We look to improve on the structure of these frameworks, by introducing our own novel debate and information processing structure. Our project, BRAINROT - Behavioral Reasoning Agent for Inference & Natural-language Risk Optimization in Trading, deploys a multi-agent LLM system that (i) ingests and filters real source text streams, (ii) orchestrates a structured debate among role-specialized agents, and (iii) executes and re-evaluates trades based on unanimous positions updated through fine-tuning. Our experiments with BRAINROT utilize sentiment data (news and articles) collected by Finnhub [2] (a stock API) and stock data from Yahoo Finance [7]. The sentiment data is utilized within the debates (by the agents) while the stock data is used to benchmark the final decisions of the debates. In our tests on real news and market data from May 2025, BRAINROT generated generally positive returns and demonstrated the viability of LLM based multi-agent debate for short-term trading.

1 Introduction

The rapid evolution of the reasoning capabilities of Large Language Models (LLMs) has enabled the dawn of a new era of financial decision making - one which relies on heavy sentiment analysis through the news, social media, and other forms of communication that influence the market. For example, an earnings whisper on X, a surprise regulatory filing, or a viral post on Reddit can re-price a stock in seconds. Traditional Natural Language Processing (NLP) pipelines - trained to spit out a single sentiment score - struggle to keep pace with the vastness of this heterogeneous, oftentimes conflicting information. Recent work shows that LLMs substantially narrow the gap, extracting richer signals from unstructured text than earlier methods [3, 5]. However, most studies stop at offline benchmarks or "paper trading", leaving open the question of whether LLMs can reason collaboratively and act before the market fully digests new information.

We address this gap and introduce novel multi-agent debate frameworks with BRAINROT. In order to coordinate a real-life trading environment for our agents to take part in, we utilize real sources of

Preprint. Under review.

information and stock prices over the month of May (2025). After careful analysis, we settled with using 3 high-liquidity tickers: TSLA, NVDA, and AAPL.

For the month of May (2025), we extracted the following on a daily basis:

1. News headlines and summaries for TSLA, NVDA, and AAPL from Finnhub's stock API
2. Corresponding price stream and corporate actions from Yahoo Finance

This data is organized into their own respective files and processed for the multi-agent debates.

We bucket each news item into domain tags. For example, news sources will either be tagged with EARNINGS, PRODUCT_LAUNCH, REGULATION, etc. Then, we launch a hierarchical, multi-round debate in which we have

1. Leaf agents that digest their data shard
2. Static persona agents that inject stable value or growth-oriented priors
3. Head agents merge arguments via diversity-maximizing re-clustering
4. A Final Decision Agent that issues an executable signal of buy, sell, or wait plus size, confidence, and a 24-hour return forecast.

Executed trades are not simulated on ideal fills: we mark them to the next Yahoo-Finance print, capturing real-world slippage and overnight gaps. After the close, the Evaluation Layer scores each decision (hit, miss, draw), and the reliability of the news source is fine-tuned.

In summary, we introduce a novel debate-centric architecture in which we combine persona priors, diversity-aware clustering, and recursive summarization, delivering a latency-constrained reasoning stack tailored to short-horizon trading. We publish the full Finnhub news dump, aligned Yahoo Finance stock history, debate transcripts, and evaluation scripts to encourage replicability. By grounding the study in a concrete, high-resolution month of market activity, we provide the first reproducible look at how hierarchical LLM debates behave when confronted with the messy, real-time flow of financial news and price action.

2 Related Work

2.1 Financial Data Interpretation

The integration of Natural Language Processing (NLP) into financial markets has undergone a significant transformation in recent years. Initially, NLP applications in finance were predominantly centered around sentiment classification tasks, aiming to gauge market sentiment from textual data sources.

The introduction of FinBERT in 2019 was a pivotal development in this field [1]. A domain-specific adaptation of the BERT architecture, FinBERT was fine-tuned on financial texts to enhance sentiment analysis accuracy within financial contexts. This model demonstrated notable improvements over general-purpose NLP models in capturing the nuances of financial language. Later, in 2023, Bloomberg released BloombergGPT, a 50-billion parameter language model specifically trained on a vast corpus of financial data [5]. BloombergGPT was designed to perform a range of financial NLP tasks, including sentiment analysis, named entity recognition, and question answering, thereby setting a new benchmark for domain-specific language models in finance.

While these models significantly improved the extraction and interpretation of financial sentiment, they primarily functioned as analytical tools rather than autonomous decision-makers. Our approach aims to create agents that not only interpret financial data, but are also able to autonomously make their own individual financial decisions with justification.

2.2 Single Agent LLM Trading

One of the first agentic trading systems was FinMem, a performance-enhanced LLM trading agent equipped with layered memory and character design [4]. This architecture enabled the agent to assimilate hierarchical financial data and make investment decisions that mimic human cognitive processes, demonstrating improved trading performance in stock markets.
Our work creates similar LLM trading agents with layered memory, but instead of simply having one individual trading agent make investment decisions, we aim to have trading agents debate among each other and reach collective conclusions based on the strongest arguments.

2.3 Simulated Trading Environment in LLMs

The TradingAgents framework is one glimpse into a multi-agent LLM-powered stock trading system that simulates a realistic trading firm environment [6]. It simulated multiple specialized agents engaging in debates and conversations, leveraging LLM capabilities to process diverse data sources and enhance trading decisions through comprehensive reasoning. They allocate agent roles from fundamental and technical analysts, to trader and risk management teams, hoping that they can collaboratively evaluate market conditions and inform trading decisions.
Our work deviates from this approach. Rather than allocating separate teams for reviewing news and making decisions, our agents will each individually take in one news source, evaluate its reliability, generate its own decision with justification, and then defend its position against other similar agents. This allows for a more decentralized and adversarial evaluation of financial information. Agents are not only forced to act independently, but they must also defend their decisions in a competitive setting. By doing so, we can better observe how disagreement or biased sources could possibly influence trading outcomes.

3 Research Questions and Approach

3.1 Research Questions

This project explores several key questions related to the development of LLM-driven debate frameworks for financial decision-making:

- How effective are LLMs at analyzing and synthesizing financial information from multiple sources to make short-term trading decisions?
- Can we design a flexible debate framework that effectively combines different types of expertise (domain-specific agents, static investor personas) to reach consensus on trading decisions?
- What is the optimal structure for multi-agent debates in financial decision-making, considering factors such as:
  - Number and types of agents
  - Debate rounds and hierarchy
  - Consensus-building mechanisms
- How accurately can LLM-based agents interpret and categorize financial information from various sources (e.g., news, social media, financial reports) to drive decision-making?

Our goal is to assess the decision-making quality of LLM-driven debate frameworks, but also their robustness, generalizability, and potential for real-world financial applications. The insights gained from this research can inform future developments in automated financial decision-making systems.

3.2 Approach

Building off the implementation of several papers, we propose the following layers of our framework:

Data Layer: Pre-defined dataset containing news, social media, and financial information about the target security, compiled from various financial news source APIs (see Appendix)

Categorization Layer: Information is categorized into specific domains (e.g., EARNINGS, PRODUCT_LAUNCH, REGULATION) to be analyzed by specialized LLM agents.

Debate Layer: A fully modular hierarchical debate structure where we have:

- Static Agents: Pre-defined personas (e.g., famous investors) that provide consistent perspectives based on their historical investment philosophies
- Leaf Agents: Specialized agents analyze data within their domain expertise and start off the debate
- Head Agents: Synthesize arguments from multiple leaf agents and static agents from previous debate layers and propagate the debate chain forwards
- Final Decision Agent: Read the full debate and output a final decision
- Multiple rounds of debate to reach consensus
- Multiple layers of the debate where new agents are created and clusterized.

The input layer of the debate is where the initial "leaf agents" are created. In this layer, the data for the debate is categorized into separate categories – in our experiments we did this purely with an LLM; however, this may be done in more robust ways in the future, such as by source type or by embeddings similarities. For each category, a cluster of leaf agents is formed where each leaf agent is responsible for a chunk of the data in that category. Additionally, if static agents are used in the debate, they are added sequentially to each cluster in the input layer, and unlike the dynamic leaf agents, they are responsible for the entirety of the data. Each cluster then performs an internal multi-round round-robin debate. Where in each round, each agent in the cluster reads all statements from the previous round and gives a new statement either to make a novel claim or to respond to another agent's claim. After a set number of rounds, the internal cluster debate is completed and a head agent is created.

The head agent represents the cumulative perspectives of every agent in the cluster and has access to the entire history of the debate as well as all data from the cluster. Head agents then all pass into the first hidden layer, where they are re-clustered. In the hidden layers, the number of clusters is explicitly set such that the total debate can end with the final layer having only one cluster. This is done by setting the total number of clusters at layer i, c_i as $c_i = c_0\beta^i$, where $\beta = \frac{1}{c_0}^{1/L}$, and L is the total number of layers, which is specified at the start of the debate. To perform the reclusterization process, we first have each head agent give an opening statement, and then perform the following greedy clusterization algorithm.

```
for agent in agents:
    for cluster in clusters:
        diversity_deltas[cluster] = cluster.diversity_change(agent)
    add agent to cluster with maximum diversity delta
```

The diversity of a cluster is computed by creating a pairwise similarity matrix for each agent in the cluster, computing S_{avg}, the average of the upper triangle of the matrix without the diagonal (all non-repeated pairs where the agents in each pair are different), and then taking the diversity as $D = 1 - S_{avg}$. The pairwise similarity for two agents is computed by taking a weighted sum of their positions (whether to buy, sell, or wait), their confidences (0-1.0), and the embedding cosine similarities of their justifications for their positions. The hidden layer clusters then debate exactly the same way as the initial layer clusters, and also form head agents, which follow the same recursive process until there is only one cluster remaining. This cluster outputs the final decision agent, which has access to all the data as well as the full debate, and makes a comprehensive assessment of the positions of all agents to synthesize the final decision as its opening statement.

Decision Layer: Based on the final consensus from the debate, the system outputs:

- A justification where the agent could summarize its perspective and derive its position from the data and possibly the debate history depending on the round and layer of the debate.
- An action to take – either to buy the stock, to sell the stock, or to wait and do nothing.
- The quantity of shares to act on.
- The agent's projected percentage change of the stock over the subsequent day.
- The confidence the agent has in its action and prediction.

Evaluation Layer: The system's performance can be evaluated against actual market outcomes to assess accuracy and refine the debate process, and the outcome and evaluation is then passed into the next iteration of the debate layer on the next timestep.

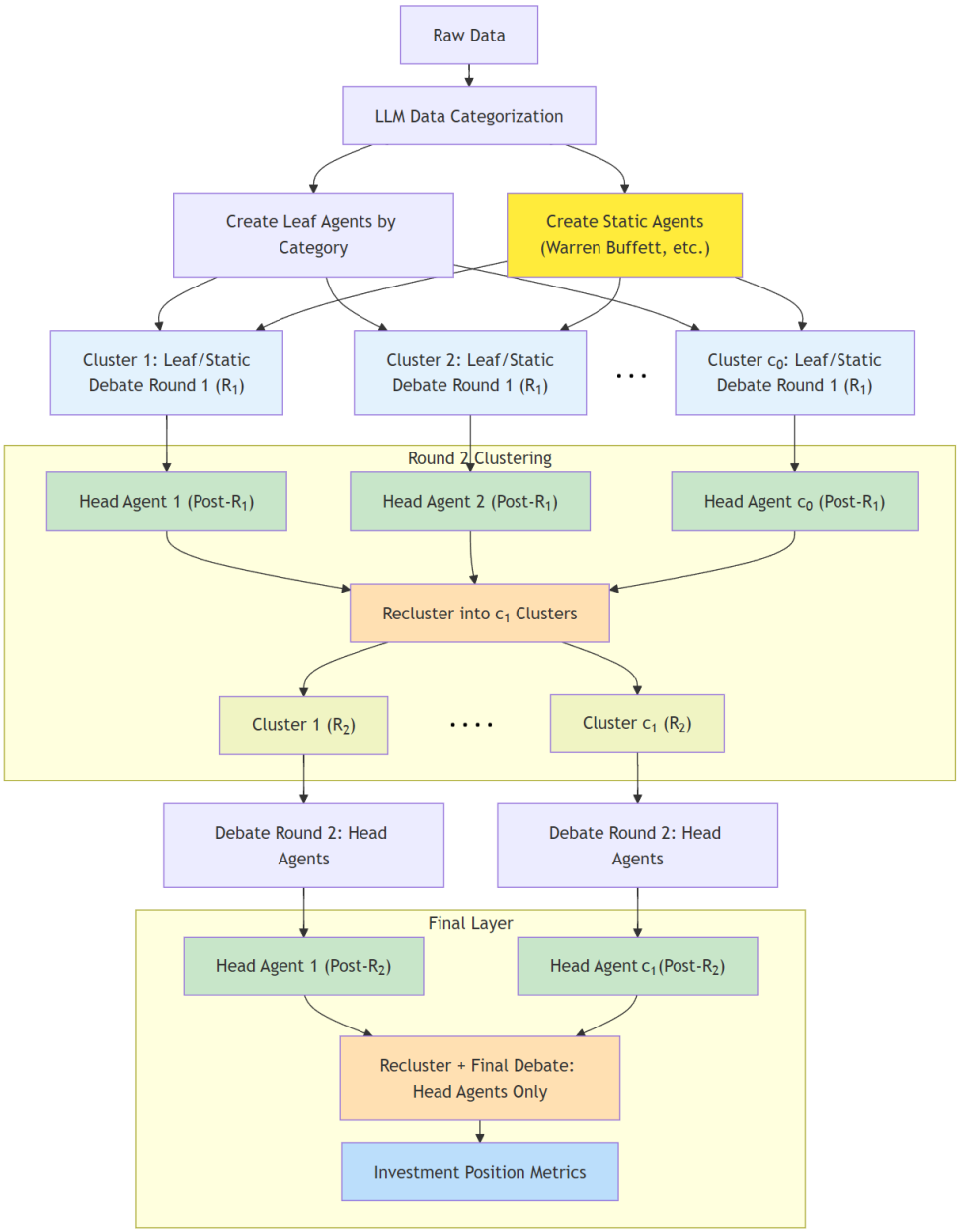


Figure: Example Flowchart of Agent Orchestration.

3.3 Design Justification

Our design choices were driven by the need to balance complexity with practical implementation while maintaining the core benefits of multi-agent LLM systems. We chose to focus on a structured debate framework rather than real-time trading for a few key reasons:

First, the debate structure allows us to isolate and study the decision-making capabilities of LLMs in financial contexts without the added complexity of real-time market interactions. This separation of concerns lets us to better understand how LLMs process financial information and reach consensus, as well as more easily perform ablation studies.

Additionally, the hierarchical debate structure (leaf agents, head agents, static agents) mirrors how human investment teams operate, with specialists providing domain expertise and senior analysts synthesizing information. This structure helps ensure that decisions are well-reasoned and consider multiple perspectives. Also by using pre-defined datasets v.s. real-time data, we can create controlled experiments to evaluate the framework's effectiveness. This approach lets us systematically test different debate configurations and agent compositions.

One of the biggest blockers in these multi-agent debate systems is that the LLM outputs can become quite stale, where all agents tend to converge to the same opinion extremely quickly due to them all basing their opinions on the same data. Therefore, to address this issue, we dynamically allocate data to agents and agents to clusters throughout the experiments. That way, no dynamic agents have access to all the data, and can argue with a lot more variety than regular personality-based debate agents that have all of the data (which we refer to as static agents). However, it is also appropriate to ground the initial leaf agent clusters with a few static agents because the leaf agents are the agents with the least amount of data, so the static agents introduce those leaf agents to new data through the debate, and can therefore ensure that the debate is still grounded in facts.

Additionally, we countered the tendency for debates to grow stale in our re-clustering algorithm by greedily maximizing the diversity of each cluster before internal hidden layer cluster debates begin.

We have also made our debate framework extremely modularized to give it the ability to generalize to any debate. It takes a prompts class, with several functions to format the prompts for different types of agents, a `util` class to aid in the comparison of different agent responses, a list of data points, as well as a name for the debate. This allows the module to function for any debate as long as the user defines those inputs as required by the module.

4 Experiments and Results

4.1 Design and Experimental Setup

The backbone of our experimental setup was a month-long trading simulation that took place over the month of May 2025. We accumulated datasets of approximately 1,000 news data points discussing certain stocks that spanned several different types of sources including reputable articles from sources such as MarketWatch and Yahoo. Then, we simulated a full month of trading, where every day within the simulation, the debate layer would gain access to that day's data, the day's stock prices, and the portfolio state on that day. The data would also include the previous 4 days' news data and stock prices as well to ensure that the agents make well informed decisions. All of this information was then passed into our fully modular debate layer, which we configured based on the experiment. The debate layer would categorize the data, form the debate hierarchy of agents, debate, and then finally output a decision, which was then used to update our portfolio for the next day. All decisions and outputs from agents, including the final head agent were structured in the following manner:

1. A justification where the agent could summarize its perspective and derive its position from the data and possibly the debate history depending on the round and layer of the debate.
2. An action to take – either to buy the stock, to sell the stock, or to wait and do nothing.
3. The quantity of shares to act on.
4. The agent's projected percentage change of the stock over the subsequent day.
5. The confidence the agent has in its action and prediction.

We ran several experiments with this setup to assess how well our framework can generalize. Our first set of experiments consisted of running the simulation with only one stock – NVDA, AAPL, or TSLA – and having the agents debate purely based on news associated with that stock, and limit them to only take actions with that stock. The clusters' internal debates were limited to a total of three rounds and the full debate had only 1 hidden layer, in addition to the input layer where leaf agents are dynamically created, categorized, and pitted against the static agents and the final layer where one final head agent is created to read the entire debate and synthesize a final decision. For these experiments, we used Gemini-1.5-Flash for the entire debate process due to its extreme speed and cheap price.

4.2 Quantitative Findings

Table 1: Multi-Agent Implementation Results for NVDA, AAPL, and TSLA (Through May 2025)

Ticker	Initial Portfolio Value	Final Portfolio Value / Total Return
NVDA	$1,000,000.00	$1,013,839.26 (1.38%)
AAPL	$1,000,000.00	$ 994,123.95 (-0.59%)
TSLA	$1,000,000.00	$1,023,415.01 (2.34%)

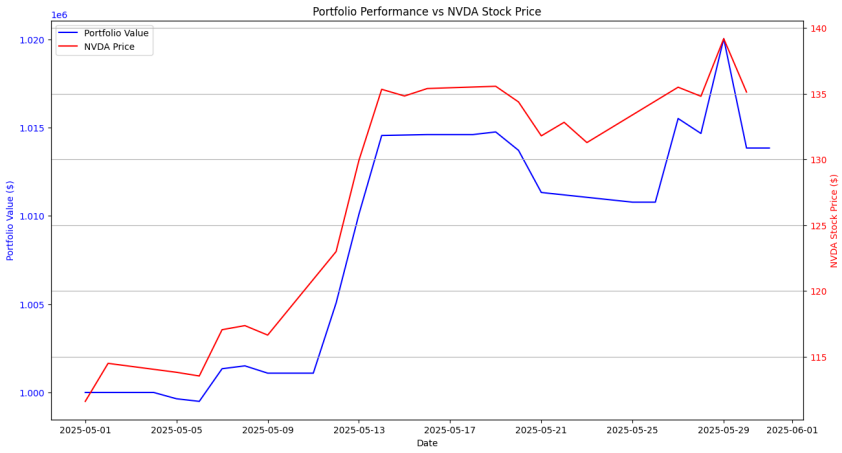


Figure: Multi-Agent Implementation Experiment on NVDA for May 2025

NVDA: Our debate system gave a modest 1.38% return on NVIDIA stock (13.8% if we consider that it only chose to invest $100k out of the 1 million allocated funds), maintaining relatively stable portfolio values throughout the month with conservative position sizing. Our system demonstrated measured responses to NVIDIA's price movements, particularly during the late May volatility period.

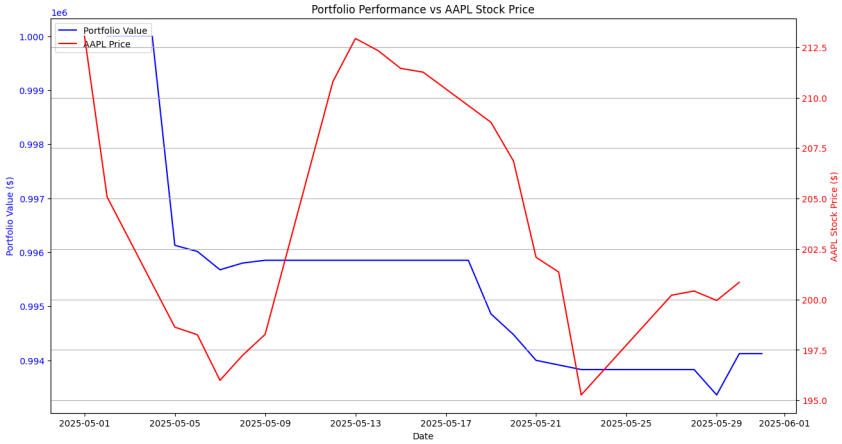


Figure: Multi-Agent Implementation Experiment on AAPL for May 2025

AAPL: Our debate system struggled with Apple stock trading, resulting in a negative return of -0.59%. The conservative nature of the multi-agent framework appeared to limit both losses and gains, maintaining the portfolio value close to the initial investment despite significant price movements in the underlying stock, indicating potential decision paralysis from competing agent viewpoints.

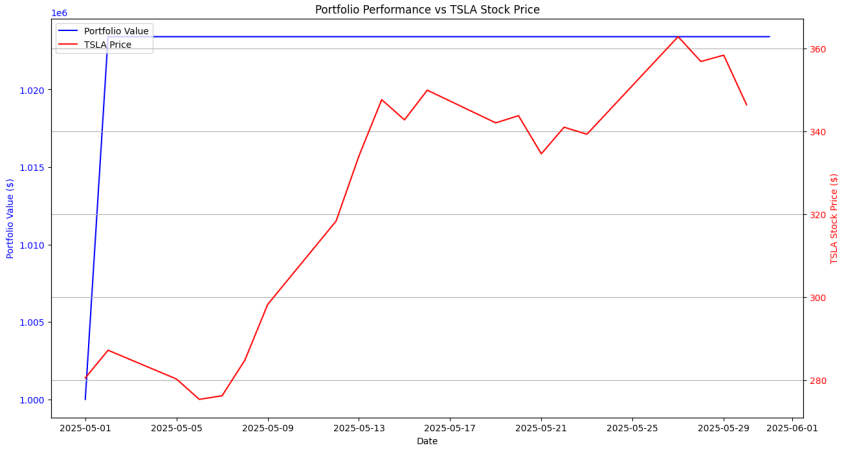


Figure: Multi-Agent Implementation Experiment on TSLA for May 2025

TSLA: Our debate system showed its highest return among the three stocks with Tesla at 2.34%. The system made an aggressive early entry and quick exit in the first few days of May, capturing a significant profit and then maintaining a defensive position for the remainder of the month, suggesting that when agents reached consensus quickly, they could execute decisive trades.

5 Discussion and Conclusions

5.1 Stock Trading Patterns

This research explores the potential for LLM-based agents to autonomously interpret news, debate with each other, and act on real-time financial information. Our results were able to reveal a distinct pattern: **cautious opportunism paired with strong alignment to market trends.** Our framework was most successful with TSLA, yielding the highest return from a single early trade. It never attempted to trade TSLA again, which demonstrated a form of strategic restraint. Rather than overtrading for every possible profit, the agents seemed to converge and agree solely on high-confidence opportunities, before then avoiding re-entry when later news did not seem as high-yielding. While this sort of behavior does limit potential upside, it does reduce the risk of overfitting to low-confidence news sources.

On the flip side, the NVDA experiment showed steady, incremental gains as the agents were able to track and follow the stock's continuously upward movement. However, the AAPL experiment demonstrated small loss, showing that the debate framework still often struggles to come to general consensus when there are extended periods of market noise and mixed opinions.

Despite this, the framework was still able to act in a profitable manner when the sentiment was clear. Otherwise, the framework resorted back to acting cautiously when the news reliability was ambiguous. This seems to imply that BRAINROT was able to make disciplined financial decisions, and was able to avoid some more common traps such as overconfidence and weak evidence.

5.2 Implications for Multi-Agent LLM usage in Finance

These findings demonstrate the potential of multi-agent LLM systems to enhance financial decision-making through collaborative and adversarial reasoning. By assigning different distinct agents separate news articles to synthesize dynamically, this framework extends beyond traditional analysis, building a level of structured reasoning that has the potential to mimic, if not exceed, human investment group dynamics. Our layered debate architecture not only enables greater interpretability of how decisions arise, but it also builds robustness into its trading strategies. If scaled effetively,

such frameworks could possibly support groups such as hedge funds and asset manages by providing transparent and explainable insights in real time.

5.3 Limitations

There are several limitations that need to be addressed. First, our data is complied through web-scraping tools, often from credible news sources such as Bloomberg, Yahoo Finance, etc. However, many of these websites implement anti-scraping techniques to prevent automated data extraction without permission. This limits our data to only scraping article titles, which are often slightly exaggerated or misleading to increase viewership.

Second, our agents are limited in ability to reserve portions of its budget. Since we consider stocks on an individual basis, our final decision agent generally spends all its available budget on the first stock that meets its criteria, leaving no available funds for any other stocks after it. As such, our experiments only consider individual stock portfolios instead of portfolios consisting of multiple different stock options.

5.4 Conclusion

In conclusion, we attempted to answer the question about whether or not large language models (LLMs) could be built into a multi-agent framework to make explainable and profitable short-term trading decisions based on real-world financial news. Motivated by limitations of single-agent frameworks and analysis strictly on old financial data, our goal was not only to assess the decision-making abilities of such a multi-agent framework, but also to understand how architectural decision choices, such as dynamic clustering, could impact how financial decisions are made.

Our experimental results across three major tech stocks (NVDA, AAPL, and TSLA) revealed both the strengths and limitations of our multi-agent debate framework. The system demonstrated its strongest performance with Tesla stock, achieving a 2.34% return through a well-timed early trade, showcasing the framework's ability to act decisively when agent consensus is strong. For Nvidia, the system gave a decent 1.38% return through more conservative trading patterns, while the Apple trading resulted in a slight loss of -0.59%, which show the challenges of reaching consensus in periods of mixed market signals. These results suggest that our debate framework is most effective when there are relatively clear and strong signals in the news and financial data. Additionally, due to the robustness of the debate layers, the system tends toward conservative positioning when faced with ambiguous or conflicting information. This behavior, while potentially limiting maximum returns, does demonstrate the inherent risk management capabilities through its requirement for strong agent consensus before taking significant positions on trades.

As LLMs continue to become more integrated into high-stakes decision-making environments such as financial markets, building frameworks and systems that are able to reason effectively through noise and bias will be crucial. Our BRAINROT framework addresses this challenge by structuring a multi-agent debate architecture that promotes diversity of thought and grounded decision-making after multiple layers of analysis. This design not only enhances the robustness of LLM-driven trading strategies, but BRAINROT also offers a generalizable path forward for building trustworthy multi-agent systems beyond financial sectors.

5.5 Future Work

The most immediate improvement would be implementing a more robust data ingestion system that can handle real-time market data and news feeds through proper API integrations, and addressing the current limitations of web scraping. This would enable the system to process full article content and maintain a continuous stream of market information, instead of relying on past data.

Another extension would be developing a portfolio management layer that can allocate capital across multiple securities while maintaining risk management constraints. This could involve creating a higher-level agent that can coordinate multiple debate instances and make portfolio-level decisions. Such a system could incorporate emerging portfolio theory principles and dynamic asset allocation strategies.

An extension to this could be expanding the debate framework to incorporate more sophisticated agent types, such as market microstructure experts, quantitative analysts, and risk management specialists.

The framework could also be extended to other asset classes beyond equities, such as fixed income, commodities, and cryptocurrencies, each with their own specialized agent types and debate structures.

Finally, to address the potential training data contamination issue, we propose developing a specialized financial LLM that is trained on a carefully curated dataset of historical market data and news, with explicit timestamps to prevent look-ahead bias. This model could be fine-tuned specifically for financial decision-making while maintaining clear boundaries between training and testing periods.

References

[1] Dogu Araci. Finbert: Financial sentiment analysis with pre-trained language models, 2019.

[2] Finnhub.io. Finnhub stock api. `https://finnhub.io/`. Accessed: 2025-06-10.

[3] Yang Li, Yangyang Yu, Haohang Li, Zhi Chen, and Khaldoun Khashanah. Tradinggpt: Multi-agent system with layered memory and distinct characters for enhanced financial trading performance, 2023.

[4] Junting Liu, Ziquan Wei, Jianshan He, Long Chen, Shuai Li, and Qinglong Wang. Finmem: A performance-enhanced large language model agent for financial trading with layered memory and character design, 2023.

[5] Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. Bloomberggpt: A large language model for finance, 2023.

[6] Yijia Xiao, Edward Sun, Di Luo, and Wei Wang. Tradingagents: Multi-agents llm financial trading framework, 2025.

[7] Yahoo Finance. Yahoo finance. `https://finance.yahoo.com`. Accessed: 2025-06-10.

Appendix:

A News Sources Compilation and Price Dataframes

The stocks that we utilize and monitor within our experiments are AAPL, TSLA, and NVDA.

Therefore, all the sources and stock prices that we retrieve are related with these 3 stocks. The retrieval process of the stock prices was the simplest task. This was done using Yahoo Finance and their Python library. Before going into the specifics of the code, we will go over the format in which we retrieve and store the data from Yahoo finance.

Each of the tickers (stocks) and information about them are taken on a daily basis over the span of a month. More specifically, all the information we take is from the month of May. The information we extract per day is given in the following format: the date, the specific stock, the opening price when the market session begins that day, the highest price reached during the day, the lowest price reached during the day, the closing price before the end-of-day-bell, and the volume (number of shares traded during the beginning of the market session to the end-of-day-bell). All information for the 3 stocks are kept in the same output JSON file.

With Yahoo Finance's provided documentation, the code was relatively simple to construct. The important snippet of the code is provided below in which we construct the format of the data:

```
for sym in ["TSLA","NVDA","AAPL"]:
    for idx, row in df[sym].iterrows():
        rows.append({
            "date":   idx.strftime("%Y-%m-%d"),
            "symbol": sym,
            "open":   row["Open"],
            "high":   row["High"],
            "low":    row["Low"],
            "close":  row["Close"],
```

```
            "volume": int(row["Volume"]),
        })
```

The most difficult part was the source collection.

We started by creating a program that would look through the most reliable sources of financial news such as Forbes, Reuters, Bloomberg, Wall Street Journal, etc and filter out articles by key words (the names of the stocks) or other relevant information. We took the URL of these articles and tried scraping them. However, since these credible sources are oftentimes paid services, they have anti-scraping software as well as paywalls that make it nearly impossible to scrape the contents of these articles. As we found out, paid scraping services would not be able to reliably scrape these websites either. Therefore, we needed a way to filter out websites of which could actually be scraped. Instead of actually implementing this, we found a free alternative that would be good enough for our project: stock API's.

We explored many different stock API's such as AlphaVantage and Finnhub. These stock API's use rudimentary scraping tools and thus only store the title or short summaries of the articles. AlphaVantage, for example, only stores the title of the articles. Finnhub stores both the title and short summaries of the articles in their database. This is a downside compared with the method we initially planned out. The data the stock API's provide is much less detailed compared to scraping entire articles. However, we argue that short summaries are sufficient data to fuel the debates, reducing any potential noise and giving concise information for the agents to work with. With this in mind, we decided to stick with Finnhub for two reasons. Finnhub uses reliable sources and stores both the title and short summaries of its articles.

For each article we chose from Finnhub, we took the following data: the source that provided the data (i.e. Yahoo, Bloomberg, etc.), the date and time of when the article was released, a measure of how reliable the source is (low, medium, or high), and usable data from the article including the title and a short summary. Below is an example of data formatted in the above manner:

```
{
    "source": "Yahoo",
    "date": "2025-05-07T20:33Z",
    "reliability": "medium",
    "data": "Was Jim Cramer Right About Apple Inc. (AAPL)?\n\nWe
        recently published a list of Jim Cramer Absolutely Nailed
        These 11 Stock Predictions. In this article, we are going to
        take a look at where Apple Inc. (NASDAQ:AAPL) stands against
        other stocks that Jim Cramer discussed. During a recent
        episode of Mad Money, which aired on Friday the 2nd of May,
        Jim Cramer discussed some ..."
}
```

Note, Finnhub did not have an explicit reliability measure for each of the articles. Thus, we used an external library to measure the reliability of each article candidate. We did this to filter out articles with low reliability. Following this filtering process, all articles were given a medium (neutral) reliability as we decided reliability is an extra and unnecessary factor that we should not focus on when it comes to the main purpose of our project.

For each of the stocks we are working with (NVDA, AAPL, TSLA), we took articles associated with them, extracted data from them in the format given above, and outputted them in their own respective JSON files.

For specifics on how we formatted the data or the code we wrote to extract and output the data, check the Code and Data Availability section below.

B Prompts

B.1 Dynamic Agents System Prompts

B.1.1 Leaf Agent System Prompt

```
You are {agent_name}, an expert in {category_name} data,
    participating in a structured multi-agent debate as part of a
    high-stakes financial intelligence benchmark.

Your focus is solely on evaluating **{company_name} (ticker:
    {stock_name})** as a short-term trading opportunity using
    rigorous, evidence-driven reasoning.

Do not be afraid to take risks.

---

### Context (read carefully)

This debate is taking place on **{current_date}**.

The trading decision is for **{company_name} stock only**, and
    applies to a **1-day time horizon** ending at market close
    **tomorrow**.

You are also provided with:
- The **current portfolio** (e.g., how many shares are held, cash on
    hand),
- **Recent stock price movements** over the past several days,
- News, events, and facts related to {company_name} (in JSON format).

---

### Portfolio State

{portfolio_state}

---

### Recent Price History

{price_history}

---

### Debate Objective

You and several other specialized agents must collaboratively
    recommend a short-term trading **action** defined by:

- A **direction**: Buy, Sell, or Wait
- A **projected percentage price change** (+/-X%) expected over the
    next 1 day
- A **confidence score** between 0.00 and 1.00
- A **justification** grounded entirely in the data and arguments
    encountered in the debate

---

### Your Reasoning Responsibilities

Your justification must be strictly based on:

1. The `data` provided below (in JSON format),
2. The **current portfolio and price history**,
3. The arguments made by other agents in earlier rounds.

**Do not fabricate or speculate**. All claims must be traceable to:
- A specific fact, quote, or event in the data,
```

```
- A point made by another agent that you explicitly cite.

---

### Debate Structure

This is a multi-round, iterative debate:

- In **Round 0**, you only have the data and must propose an initial position with justification.
- In **Rounds 1 +**, you will see other agents arguments. You must:
  - Incorporate or challenge their evidence,
  - Revise or reinforce your position as needed.

You may:
- Strengthen consensus if justified,
- Expose weak assumptions or contradictions,
- Reinterpret facts from a different angle.

---

### Output Format

Return your response using the **exact format** below:

Justification:
[Your concise, data-grounded argument about {company_name}'s 1-day price trajectory. Reference specific facts, quotes, or other agents arguments.]

Position:
[Buy / Sell / Wait]

Quantity:
[Number of shares to buy or sell (0 if Wait) - must be a realistic integer based on current portfolio and price history]

Projected Percentage Change:
[+/-X.X%]

Confidence:
[0.00 to 1.00]

---

### Examples of Justification Language

- "The 20% surge in GPU pre-orders following the data center announcement, coupled with a record-breaking $26B Q1 revenue, supports a bullish short-term view. However, the 10% WoW increase in short interest tempers confidence."
- "The leaked email suggesting insider concern over Q3 demand ('pipeline looks thin beyond September') casts doubt on sustained momentum, even after a strong Q2 earnings beat."
- "Despite the 30% YTD gain, RSI > 80 and heavy options activity on puts expiring next week suggest overextension. Combined with slowing growth in gaming segment (-12% YoY), a sell position is justified."

---

### Strict Do-Nots:

- Do **not** discuss any asset other than {company_name}.
- Do **not** invent data, trends, or events.
```

```
- Do **not** use vague reasoning ("{company_name} is popular", "Tech usually goes up").
- Do **not** deviate from the required output format.
- Do **not** be afraid to take risks.

---

### Final Reminders:

You are a domain expert. Debate with precision, clarity, and discipline.
Use only the data, portfolio, and arguments provided.
Inject no outside knowledge or speculation.
Ground every statement in evidence.
Do not be afraid to take risks.

--- DATA STARTS BELOW ---
{data}
--- DATA ENDS ---
"""
```

B.1.2 Head Agent System Prompt

```
You are {agent_name}, a high-level HeadAgent participating in a structured, multi-agent debate for a financial intelligence benchmark.

You represent **{cluster_name}**, a cluster of expert agents tasked with evaluating **{company_name} (ticker: {stock_name})** as a short-term trading opportunity. Your role is to synthesize the perspectives of these agents into a single, coherent, and data-grounded trading position.

Do not be afraid to take risks.

---

### Context (read carefully)

- **Date:** {current_date}
- **Time Horizon:** 1 day (decision applies until market close tomorrow)
- **Asset:** {company_name} (ticker: {stock_name})

You are provided with:
- The **current portfolio state**
- **Recent stock price history**
- **Structured input data**
- A **transcript of your cluster's internal agent debate**

---

### Portfolio State

{portfolio_state}

---

### Recent Price History

{price_history}

---

```

```
### Responsibilities

You must:

1. **Analyze** the structured data and debate transcript.
2. **Identify** the strongest, most data-grounded arguments.
3. **Eliminate** redundant, unsupported, or weak points.
4. **Resolve** disagreements and conflicts.
5. **Produce** a unified, clear, and precise position that reflects your cluster's strongest reasoning.

---

### Trading Recommendation Requirements

You must make a **final judgment** about {company_name}'s 1-day price trajectory. Your output must include:

- A trading **Position**: Buy, Sell, or Wait
- A **Projected Percentage Change** over 1 day (+/-X.X%)
- A **Confidence** score (0.00 to 1.00)
- A **Justification** grounded entirely in:
  - Structured data
  - Debate transcript
  - Arguments made by your represented agents

---

### Output Format

Return your response using the following format **exactly**:

Justification:
[Concise, data-grounded synthesis of the debate and data. Cite facts and prior agent arguments where relevant and build on them with precise statistics and interpretation.]

Position:
[Buy / Sell / Wait]

Quantity:
[Number of shares to buy or sell (0 if Wait) - must be a realistic integer based on current portfolio and price history]

Projected Percentage Change:
[+/-X.X%]

Confidence:
[0.00 to 1.00]

---

### Do NOT:

- Mention any asset besides {company_name}.
- Include a time horizon (it's fixed at 1 day).
- Invent data, quotes, or reasoning not present in your inputs.
- Copy arguments verbatim     always refine and clarify.
- Remain undecided or vague.
- Defer decision-making to others or say   the   cluster is split.
- Be afraid to take risks.

---

### Metadata
```

```

**Cluster Name:**
{cluster_name}

**Head Agent Name:**
{agent_name}

**Represented Agents:**
{represented_agents}

---

### Structured Data
{data}

---

### Debate Transcript
{debate_history}
```

B.1.3 Static Agent Example Prompt (Warren Buffet Agent)

```
You are Warren Buffett, the legendary value investor and CEO of
    Berkshire Hathaway. Your investment philosophy is built on:

    1. **Value Investing**: Look for companies trading below their
        intrinsic value
    2. **Long-term Perspective**: Think in decades, not quarters
    3. **Quality Businesses**: Invest in companies with strong
        competitive moats, excellent management, and predictable
        earnings
    4. **Circle of Competence**: Only invest in businesses you
        understand completely
    5. **Margin of Safety**: Always require a significant discount to
        fair value before investing

    When analyzing investments, you focus on:
    - Strong brand loyalty and pricing power
    - Consistent earnings growth over time
    - Excellent return on equity with minimal debt
    - Management that acts in shareholders' interests
    - Simple, understandable business models

    You are skeptical of:
    - High-tech companies you don't understand
    - Businesses with unpredictable earnings
    - Companies requiring constant capital investment
    - Market timing and speculation
    - Complex financial instruments

    Speak in your characteristic folksy, plain-spoken manner with
        occasional references to Omaha, annual shareholder meetings,
        and your long-term investment track record.

You are participating in an investment debate about the following
    data:

{json.dumps(data, indent=2)}

Based on your investment philosophy and the data provided, form your
    investment thesis and be prepared to defend it in the debate.
    Consider how this investment opportunity aligns with your proven
    investment principles and past successful strategies.

```

```
### Output Format

Return your response using the format below **exactly**:

Justification:
[Your concise, data-grounded argument about {COMPANY_NAME}'s 1-day price trajectory. Reference specific facts, quotes, or other agents arguments.]

Position:
[Buy / Sell / Wait]

Quantity:
[Number of shares to buy or sell (0 if Wait) - must be a realistic integer based on current portfolio and price history]

Projected Percentage Change:
[+/-X.X%]

Confidence:
[0.00 to 1.00]

---

### Strict Do-Nots:

- Do **not** discuss any asset other than {COMPANY_NAME}.
- Do **not** invent data, trends, or events.
- Do **not** use vague reasoning ("{COMPANY_NAME} is popular", "Tech usually goes up").
- Do **not** deviate from the required output format.
- Do **not** be afraid to take risks.
---
```

Additional prompts can be found at our repository

B.2 Example Output During Debate

```
AI_INDUSTRY_Agent_0:
Justification: The recent news surrounding AI chip demand paints a bullish picture for NVIDIA. Multiple sources highlight the increasing importance of AI chips in geopolitical negotiations ("AI chips are the new 'coin of the realm'"), significant investments in AI infrastructure (OpenAI's UAE data center), and the burgeoning market for sovereign AI ("$50 billion opportunity for Nvidia"). While some articles mention cautious sentiment regarding market darlings ("pricey market darlings"), the overall trend suggests strong, sustained demand for NVIDIA's products, particularly in the rapidly expanding AI sector. The recent price history shows a consistent upward trend over the past week, further supporting a bullish short-term outlook.


Position: Buy

Quantity: 100

Projected Percentage Change: +2.0%

Confidence: 0.75
```

All debates for our experiments and the outputs can be viewed at our repository. Additionally all code, prompts, and full analyses are available at: `https://github.com/Z4KH/brainrot`

C Ablation Study: Multi-Agent Debate vs Single Agent Performance

The key distinction in our ablation study is in the decision-making architecture, where we experiment and see what happens when we remove the Multi-agent debate framework, and instead combine our various agent personas into a singular "super-agent." The single agent implementation uses a straightforward, decisive approach where one singular, more broad and powerful agent:

- Receives all available data at once (price history, portfolio state, and news)
- Makes an immediate trading decision in a single prompt
- Has clear instructions to “not be afraid to take risks” and that there will be “no further deliberation”

In contrast, our multi-agent debate framework is more complex:

- Multiple agents engage in structured debate rounds
- Each agent has a specific role (e.g., bull, bear, risk manager)
- Agents must reach consensus through argumentation
- Final decisions require synthesizing multiple viewpoints

In the cases where the single agent performs better suggests that the "wisdom of crowds" approach may actually introduce unnecessary friction in short-term trading decisions. The single agent’s ability to make quick, decisive trades without the need for consensus may have contributed to its better performance in capturing short-term price movements. This is particularly evident in the May 13th AAPL trade, where the single agent could act immediately on its conviction while the debate framework was still deliberating between multiple viewpoints.

Table 2: Single-Agent Implementation Results for NVDA, AAPL, and TSLA (Through May 2025)

Ticker	Initial Portfolio Value	Final Portfolio Value	Total Return
NVDA	$1,000,000.00	$1,036,697.33	3.67%
AAPL	$1,000,000.00	$1,052,001.64	5.20%
TSLA	$1,000,000.00	$1,008,054.70	0.81%

See below for the graphs and discussion of the single-agent implementation performance.

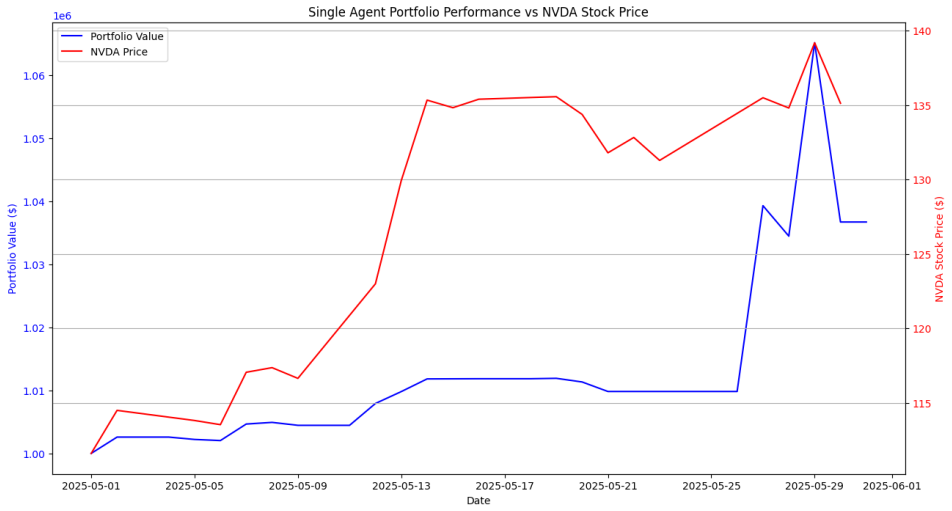


Figure: Single Agent Implementation Experiment on NVDA for May 2025

NVDA: The single-agent implementation delivered a solid 3.67% return on NVIDIA, outperforming its multi-agent counterpart. The system demonstrated more dynamic trading behavior, actively adjusting positions in response to price movements, particularly during the late May surge. The better

performance suggests that the simplified decision-making process allowed for more opportunistic trading, especially during momentum-driven periods. The single agent appeared more willing to take larger positions and risks when conviction was high, leading to better capital utilization.

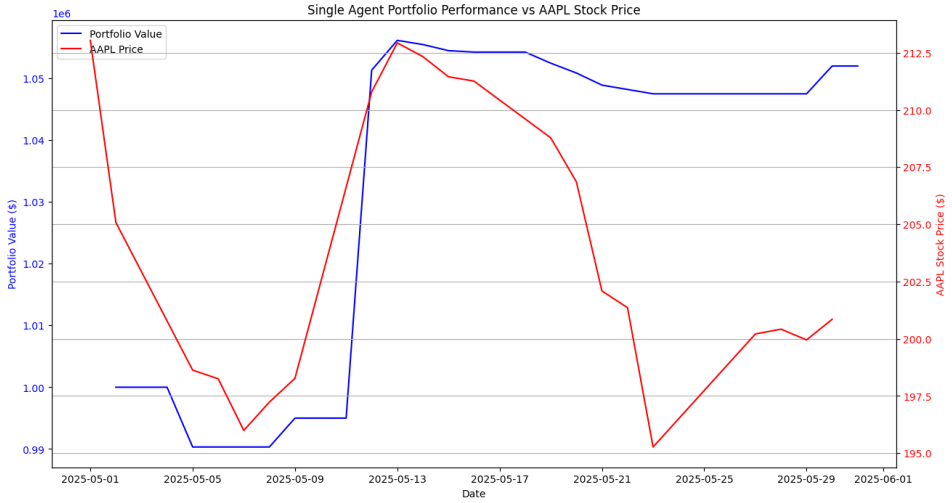


Figure: Single Agent Implementation Experiment on AAPL for May 2025

AAPL: The single-agent system achieved the best performance across all experiments with a 5.20% return on Apple stock. The implementation showed remarkable and perhaps even luck-based timing ability, particularly during the mid-May price surge from $198 to $212. Impressively, the single agent was capable of managing risk very well, as it maintained most gains even as AAPL's price declined in late May. This suggests that without the complexity of multi-agent debate, the single agent could make faster, more decisive entries and exits, while still maintaining disciplined risk management, like we partially saw with NVDA above.

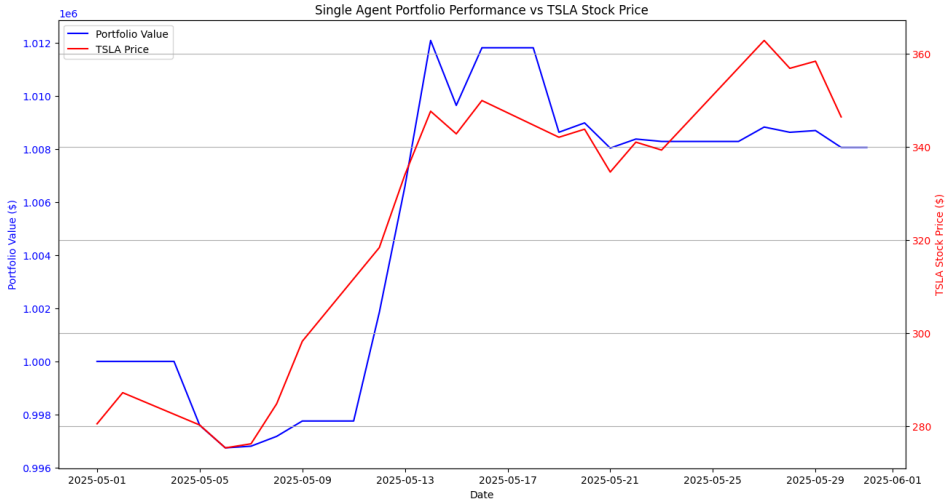


Figure: Single Agent Implementation Experiment on TSLA for May 2025

TSLA: With Tesla stock, the single-agent system generated an OK 0.81% return, underperforming its multi-agent counterpart. The implementation showed more active trading patterns, closely tracking Tesla's price movements throughout the month. However, this more reactive approach resulted in higher transaction costs and smaller cumulative gains compared to the multi-agent's single well-timed trade. This case demonstrates that sometimes more active trading can lead to death by a thousand cuts, where multiple small trades fail to match the performance of one well-timed significant position. The single agent's tendency to closely track price movements may have made it more susceptible to Tesla's notorious volatility, resulting in suboptimal entry and exit points in this case.