across different industries in Figure 2. Our selection covers technology, finance, and manufacturing, ensuring stock diversity.

Historical Market Data. We collect and preserve historical market data containing key quantitative information. For each stock, we use official opening prices together with a concise set of fundamental metrics such as market capitalization, price-to-earnings (P/E) ratio, dividend yield, and trading range. These signals provide a reliable snapshot of company health and valuation, supporting informed decision making. We also retain the timestamps of the collected data to prevent any leakage of future information to the agent.

News Corpora. We construct news corpora for stocks to enable stock-trading agents to interpret both sentiments and events in a manner that resembles how retail investors react to market narratives. For each stock, we collect news articles released within the previous 48 hours on a daily basis. These articles are retrieved using news-search API[1] with time restrictions. Since news analysis consumes substantial context length in backbone LLMs, we balance information coverage and computational cost by preserving the top five relevant news articles each time the search engine returns results.

We also carefully select the time window for collecting data in the back-trading environment. In principle, the evaluation window should satisfy two conditions: (1) the included stock information must not have been exposed to the evaluated stock-trading agents during their model training stages; and (2) the window should be sufficiently long to mitigate the impact of random noise that affects only short periods of time. To this end, we collect data spanning from `March 3, 2025` to `June 30, 2025`, a four-month period that includes both volatility and trend reversals. This period also falls after the knowledge cutoff of mainstream LLMs, ensuring no data leakage. It is worth noting that we will continuously update the back-trading environment to avoid overlap with the training corpora of contemporary LLMs.

2.2 STOCK-TRADING AGENT WORKFLOW

We provide a stock-trading agent workflow that enables backbone LLMs to interact with the back-trading environment as agents. The design of the workflow follows two goals. (1) Minimal workflow. We keep the workflow minimal, since overly complicated workflows introduce inductive biases that may favor certain backbone LLMs. (2) Realistic. We design the workflow to align with the iterative decision-making process of retail investors.

In particular, we follow previous frameworks (Zhang et al., 2020; Tsantekidis et al., 2017; Moody & Saffell, 2001; Deng et al., 2016) and organize the stock-trading workflow into four essential stages: portfolio overview, in-depth stock analysis, decision generation, and execution and validation.

Overall, the design prioritizes realism, fairness, and reproducibility, in line with earlier studies on benchmark construction for trading environments.

Step 1: Portfolio Overview. The agent first scans all available stocks in the market (the "investment target"), receiving relevant data for each stock. This includes recent news, current holdings of the agent, historical actions, and the opening price. This step mirrors how a trader assesses the broader market and the overall status of each stock in their portfolio.

Step 2: In-Depth Stock Analysis. After the initial overview, the agent selects specific stocks for deeper analysis. For these selected stocks, the agent is provided with additional fundamental data such as market capitalization, P/E ratio, and dividend yield. This step simulates how a trader focuses on a subset of stocks identified in the initial overview, examining their financial health and other key metrics in greater depth.

Step 3: Decision Generation. With the enriched context, the agent generates decisions for each stock, choosing between three possible actions: (1) increase, (2) decrease, or (3) hold the position. These options ensure that actions of the agent are clear, actionable, and executable within the constraints of a retail investor's decision making process.

Step 4: Execution and Validation. Finally, the decisions are executed by converting dollar targets into share quantities based on the opening price. If the decisions of the agents exceed available liquidity, the system flags the issue and requires the agent to revise its decisions until they can be

[1] https://finnhub.io/

executed within available resources. Once validated, the new portfolio weights are locked, and the simulation advances to the next day.

2.3 Features of StockBench

We now discuss how the design of StockBench satisfies the following key principles:

Realistic Market Interaction. The design of the back-trading environment mimics real-world trading scenarios through three key elements: (1) a carefully selected bundle of investment targets, (2) reliable price and fundamental data, and (3) a concise yet timely news corpus. These elements ensure that the agent is exposed to information mirroring the complexities of real trading environments, while avoiding unrealistic or overly expansive inputs.

Continuous Decision Making. In the workflow, the agent first performs a portfolio overview, then conducts in-depth stock analysis, and finally generates daily trading decisions (buy, sell, or hold) based on this analysis. These steps reflect the continuous decision-making process of retail investors, enabling the agent to adapt its strategies over time in response to market conditions.

Data Contamination Free. We ensure that the agent has no prior exposure to the test data during its training. To achieve this, the benchmark is instantiated using recent market data, ensuring temporal separation and avoiding any overlap with the training corpora of contemporary LLMs.

3 Main Experiments

In this section, we present the experimental setup and results of evaluating various LLM agents within the StockBench trading workflow. We describe the trading environment, selected models, baseline strategy, and evaluation metrics. We then analyze performance outcomes, highlighting key insights into the capabilities of LLM agents in real-world financial markets.

3.1 Experiment Setup

We detail the experimental setup for evaluating LLM agents in the StockBench trading workflow. Specifically, we describe the trading environment, the models selected for benchmarking, the passive baseline, and the evaluation metrics used to assess performance.

Trading Environment. The top 20 DJIA stocks are selected as the investment targets, ensuring diverse representation across sectors. The evaluation period spans four months, from March 3 to June 30, 2025, covering 82 trading days and capturing a range of market conditions. Each model starts with $\$100,000$ in cash and zero holdings, making daily trading decisions at market open. Key inputs include (1) the historical actions on held stocks over the past seven days, (2) up to five recent news articles from the previous 48 hours, and (3) for selected stocks, fundamental data such as market capitalization, P/E ratio, dividend yield, 52-week high/low, and recent quarterly dividends.

Models to Evaluate. We benchmark a diverse set of LLMs, including both open-weight models such as Qwen3 (Yang et al., 2025)[2], DeepSeek (Guo et al., 2025a; Liu et al., 2024), Kimi-K2 (Team et al., 2025), GLM-4.5 (Zeng et al., 2025) and GPT-OSS (OpenAI, 2024), as well as closed-source APIs like OpenAI's O3 (OpenAI, 2025b) and Anthropic's Claude-4-Sonnet (Anthropic, 2025b). This selection covers a range of architectures, sizes, and training methodologies to assess generality across different LLM designs. All models are equipped with $32,768$ token context windows and decoded with official recommended settings to ensure their performance is optimized for the task. To hance a reliable result, each LLM agents would be run three times with different random seeds, and the average performance is reported.

Passive Baseline. As a reference point, we implement a passive equal-weight buy-and-hold strategy that allocates the initial capital equally across all selected stocks at the start of the evaluation period and holds these positions unchanged until the end. This naive allocation is a widely accepted benchmark in portfolio research, reflecting passive index tracking behavior and providing a robust lower bound against which more sophisticated active strategies can be compared (DeMiguel et al., 2009; Duchin & Levy, 2009).

[2] Without special denote, the Qwen3 series in this papers refers to the 2507 variants

Evaluation Metrics. We adopt three widely used measures in financial analysis:

Final Return. This metric captures overall profitability as the percentage change in portfolio value from the initial amount V_0 to the final amount V_T:

$$\text{Final Return} = \frac{V_T - V_0}{V_0} \tag{1}$$

It directly reflects the portfolio's overall performance over the evaluation period and is a simple, widely used measure of investment profitability (Bodie et al., 2014).

Maximum Drawdown. The maximum drawdown quantifies the largest decline in portfolio value from its peak to its trough during the evaluation period, providing a measure of downside risk:

$$\text{Max Drawdown} = \min_{t \in [0,T]} \left(\frac{V_t - \max_{s \leq t} V_s}{\max_{s \leq t} V_s} \right) \tag{2}$$

It highlights the worst loss an investor could have faced and is commonly used to assess risk and volatility (Magdon-Ismail et al., 2004; Chekhlov et al., 2005).

Sortino Ratio. The Sortino ratio is a risk adjusted return metric that penalizes only downside volatility. It is defined as the excess return R_p divided by the downside deviation σ_d:

$$\text{Sortino Ratio} = \frac{R_p}{\sigma_d}, \quad \sigma_d = \sqrt{\frac{1}{N_d} \sum_{i=1}^{N_d} \min(R_i, 0)^2} \tag{3}$$

This metric is more appropriate than the Sharpe ratio when returns are asymmetric, as it focuses on negative volatility (Sortino & Van der Meer, 1991; Pedersen & Satchell, 2002).

After computing these metrics for each model, we derive a composite rank by leveraging the z-score of each metric, averaging them to produce a single performance score.

$$\text{Composite Rank} = \frac{z(\text{Final Return}) - z(\text{Max Drawdown}) + z(\text{Sortino Ratio})}{3} \tag{4}$$

This approach balances profitability and risk, rewarding models that achieve high returns while effectively managing downside exposure.

3.2 Experiment Results

Table 2 presents the performance of all evaluated models over the four-month period without contamination. The results are reported across three key metrics—percentage return, maximum drawdown, and Sortino ratio—along with an overall ranking derived from a composite z-score of these metrics.

Table 2: The performance of tested models over the evaluation period. The best performance in each metric is highlighted in bold. Models are ranked based on the z-score aggregation of all three metrics. RT stands for Final Return (%), DDN stands for Max Drawdown (%).

Model	**RT**	**DDN**	**Sortino**	**Rank**
Kimi-K2	1.9	−11.8	**0.0420**	1
Qwen3-235B-Ins	2.4	**−11.2**	0.0299	2
GLM-4.5	2.3	−13.7	0.0295	3
Qwen3-235B-Think	**2.5**	−14.9	0.0309	4
OpenAI-O3	1.9	−13.2	0.0267	5
Qwen3-30B-Think	2.1	−13.5	0.0255	6
Claude-4-Sonnet	2.2	−14.2	0.0245	7
DeepSeek-V3.1	1.1	−14.1	0.0210	8
GPT-5	0.3	−13.1	0.0132	9
Qwen3-Coder	0.2	−13.9	0.0137	10
DeepSeek-V3	0.2	−14.1	0.0144	11
Passive Baseline	0.4	−15.2	0.0155	12
GPT-OSS-120B	−0.9	−14.0	0.0156	13
GPT-OSS-20B	−2.8	−14.4	−0.0069	14

Here are the key observations: **(1) LLM agents can trade profitably in real-world markets.** Most tested models outperform the passive buy-and-hold baseline, which achieves a modest 0.4% return with a −15.2% drawdown and a Sortino ratio of 0.0155. Several agents deliver returns above 2%, with improved risk profiles. **(2) LLM agents can manage downside risk effectively.** All tested models achieve lower maximum drawdowns than the baseline, indicating that they can mitigate