AlphaQuanter: An End-to-End Tool-Orchestrated Agentic Reinforcement Learning Framework for Stock Trading

Zheye Deng **Jiashu Wang**
HKUST
{zdengah, jwangjv}@cse.ust.hk

https://alphaquanter.github.io/
https://github.com/AlphaQuanter/AlphaQuanter

Abstract

While Large Language Model (LLM) agents show promise in automated trading, they still face critical limitations. Prominent multi-agent frameworks often suffer from inefficiency, produce inconsistent signals, and lack the end-to-end optimization required to learn a coherent strategy from market feedback. To address this, we introduce **AlphaQuanter**, a single-agent framework that uses reinforcement learning (RL) to learn a dynamic policy over a transparent, tool-augmented decision workflow, which empowers a single agent to *autonomously orchestrate tools* and *proactively acquire information* on demand, establishing a transparent and auditable reasoning process. Extensive experiments demonstrate that AlphaQuanter achieves state-of-the-art performance on key financial metrics. Moreover, its interpretable reasoning reveals sophisticated strategies, offering novel and valuable insights for human traders. Our code for data acquisition and agent training is publicly available at: https://github.com/AlphaQuanter/AlphaQuanter.

1 Introduction

The exploration of automated trading systems in modern financial markets is flourishing. Traditional machine learning methods (such as SVM, Random Forests, etc.) [1–3] typically simplify the problem into discrete predictions of price direction at the next moment, making it difficult to effectively integrate multi-source heterogeneous trading signals. Although Deep Reinforcement Learning (DRL) can directly optimize decisions around long-term portfolio returns [4, 5], its black-box nature leads to trading decisions that lack interpretability and traceability. Recently, Large Language Models (LLMs) have demonstrated tremendous potential in the field of financial trading [6–8].

However, existing LLM-based attempts still face critical challenges. (1) First, *the lack of tool orchestration and active information acquisition capabilities* makes it difficult for models to autonomously invoke and sequentially utilize external tools during the reasoning process, identify information gaps, and fill them on demand. (2) Second, *decision transparency and interpretability are insufficient*; current training paradigms are mostly black-box end-to-end optimization or offline answer fitting, lacking a transparent display of decision reasoning trajectories and visual analysis of decision patterns, making it difficult to establish user trust and support regulatory audits. (3) Last but not least, *prompt-based methods exhibit poor robustness*, are extremely sensitive to prompt engineering, and lack effective coordination and constraint mechanisms in multi-agent debate scenarios, frequently leading to low decision efficiency, system fragility, and signal inconsistency. Overall, conducting reasoning under partially observable conditions, integrating heterogeneous signals, and executing actions with calibrated confidence remain core challenges that urgently need to be addressed.

Preprint.

To address these gaps, we propose **AlphaQuanter**, a single agent trading framework designed to enable *informative*, *explainable* and *robust* trading decisions. First, AlphaQuanter unifies the workflows into one ReAct-like agent [9] tailored for trading-oriented planning and reasoning. We define several tools for various information sources and our framework starts from a guided plan followed by iterative tool use and information seeking as well as in-depth analysis. Second, to further enhance decision making capabilities and improve model transparency, we leverage reinforcement learning with verifiable rewards [10, 11] to end-to-end optimize models that can selectively invoke useful tools and effectively gather supporting evidence. We further curate high-quality outcome- and process-based reward signals to guide RL training across diverse actions. This design eliminates the need for extensive prompt engineering across multiple agents, while ensuring both explainability and flexibility in the final decision-making process. Finally we evaluate our framework through comprehensive backtesting protocols. Our key contributions are summarized as follows:

- We propose a novel single-agent framework with effective reasoning chains that ensure both decision consistency and interpretability.
- We design an end-to-end reinforcement learning approach that trains the agent to actively acquire useful information and select evidences for in-depth analysis. It directly optimizes the entire decision-making process for long-term profitability.
- Our extensive empirical evaluations demonstrate that AlphaQuanter not only achieves state-of-the-art performance on key financial metrics, but also learns sophisticated strategies that offer practical insight for human experts.

2 Related Work

Early approaches use traditional machine learning methods, such as SVM and random forest, to frame the task as a simple price direction classification [1–3], which has been proven insufficient due to oversimplification and poor generalization in trading environments [12].

Deep Reinforcement Learning Moody and Saffell [4] pioneered the application of deep reinforcement learning to stock trading, directly optimizing trading performance end-to-end and outperforming supervised learning in long-horizon S&P 500 backtests. iRDPG [13] integrates imitation learning under a partially observable Markov decision process framework, using expert behavior to stabilize the training process and improve robustness, but overly relies on existing strategies. DeepTrader [5] introduces macro states and risk-sensitive rewards, achieving dynamic adjustment of long-short positions and risk control. MTS [14] improves returns across multiple datasets through time-aware encoding, parallel short selling, and CVaR-based risk management. However, these methods belong to end-to-end black-box optimization, lacking necessary interpretability, and cannot integrate external signals such as news and fundamentals on demand.

LLMs-Based Trading Agents TradingAgents [6] introduces a multi-agent framework that forms trading decisions through the collaborative interaction of LLM agents simulating analysts, traders, risk controllers and other roles, with backtesting results outperforming multiple benchmark methods. FinAgent [7] combines multimodal information fusion with tool enhancement, achieving state-of-the-art performance across six evaluation metrics. However, both approaches lack explicit coordination and constraint mechanisms, making debate-style decision processes yield inconsistent or conflicting signals, and high sensitivity to prompt design. Alpha-GPT [8] adopts a human-in-the-loop paradigm that enables factor mining through natural-language interaction, but it is difficult to autonomously scale and automate in high-frequency trading environments.

LLM-Based Reinforcement Learning Optimization Motivated by the recent success of DeepSeek-R1 [10], growing work explores RL approaches to optimize LLMs for quantitative trading. FLAG-Trader [15] employs partially fine-tuned LLMs as policy networks, optimizing trading rewards through policy gradient methods. Trading-R1 [16] constructs large-scale financial corpora and implements a three-stage curriculum learning framework that combines SFT with RL. However, both types of methods generally lack end-to-end simulation of real trading processes and autonomous exploration capabilities and have not yet endowed models with spontaneous perception of information gaps or proactive orchestration of external tools.

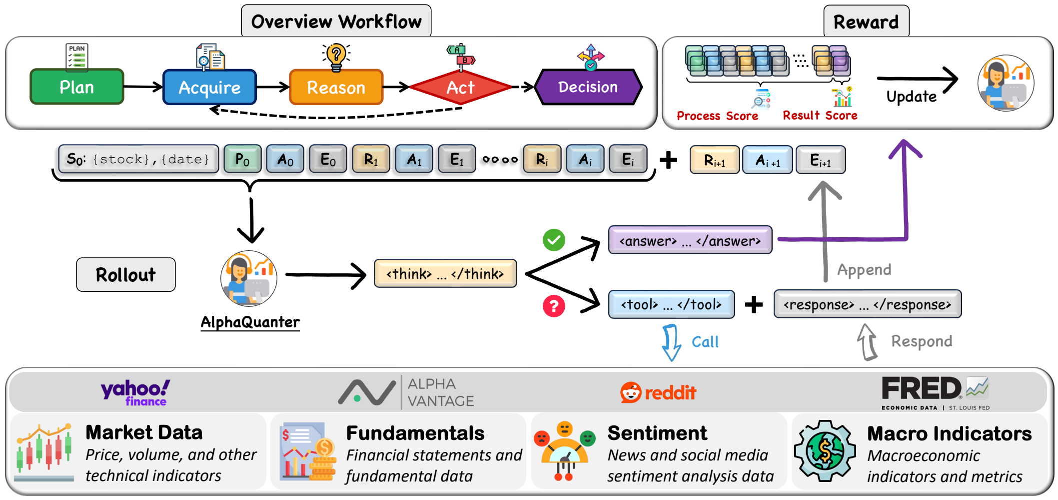

Figure 1: The overall architecture and workflow of AlphaQuanter. The central panel shows the agent's iterative rollout process. Starting from an initial state (S_0), the agent first forms an initial plan (P_0) before generating further reasoning traces (R_{i+1}) with <think> tag. In each step, it decides whether to continue acquiring information by executing a tool-based action (A_{i+1}) and receiving its environmental feedback (E_{i+1}), or to conclude by outputting a final decision with an <answer> tag. Throughout this process, the agent can query multi-dimensional financial data sources (*bottom* panel, Section 6.2). Once a decision is made, the entire trajectory will be evaluated to compute a reward (*top-right* panel, Section 4.2), which updates the agent's policy. The overall workflow (*top-left* panel, Section 4.1) is designed to mimic a human trader's cognitive process of reasoning and acquiring data on demand.

3 Problem Definition

To navigate a partially observable market within a single trading day, we model the agent's task as a tool-augmented Markov Decision Process (MDP), defined by the tuple $\langle \mathcal{S}, \mathcal{A}, \mathcal{T}, \mathcal{R} \rangle$. The central challenge within this framework is to learn a strategy that sequences tool use and the final action to maximize return.

State Space $\mathcal{S}$ A state $s \in \mathcal{S}$ captures the agent's accumulated information, represented as the tuple $s = (\texttt{initial\_context}, \texttt{query\_history}, \texttt{query\_result})$, where `initial_context` includes basic metadata (e.g., stock symbol, date), `query_history` records the tools invoked so far, and `query_result` stores their corresponding outputs.

Action Space $\mathcal{A}$ The action space $\mathcal{A}$ comprises two distinct types. First, the agent can execute a *query action* from $\mathcal{A}_q = \{f_1, f_2, \cdots, f_{|\mathcal{A}_q|}\}$ to actively gather information from four source categories (market data, fundamental indicators, sentiment analysis, and macroeconomic metrics), thereby updating its state. Detailed descriptions of the data sources are provided in Section 6.2. Finally, the agent can execute a *decision action* from $\mathcal{A}_d = \{\texttt{BUY}, \texttt{SELL}, \texttt{HOLD}\}$, which terminates the decision-making process.

Transition Dynamics $\mathcal{T}$ The state transitions are deterministic. When the agent selects a *query action* $a_t \in \mathcal{A}_q$ at time step t, the current state s_t transitions to s_{t+1} by appending the query and its result to `query_history` and `query_results`, respectively. Conversely, when the agent selects a *decision action* $a_t \in \mathcal{A}_d$, the episode terminates immediately.

Reward Function $\mathcal{R}$ An episode yields a trajectory $\tau = (s_0, a_0, s_1, a_1, \cdots, s_T, a_T)$, a sequence of states and actions beginning at the initial state s_0 and terminating with the first *decision action* a_T, where all intermediate actions $a_0, a_1, \cdots, a_{T-1}$ are *query actions*. The agent's objective is to learn a policy π that maximizes the cumulative trajectory reward $R(\tau) = \sum_{t=0}^{T} \mathcal{R}(s_t, a_t)$. The step-wise reward function $\mathcal{R} : \mathcal{S} \times \mathcal{A} \to \mathbb{R}$ is designed to promote strategic and profitable decision-making:

rewarding BUY when the outlook is positive, SELL when negative, and HOLD when conditions are neutral or non-directional, while guiding tool use toward informative queries.

4 AlphaQuanter

4.1 Cognitive Workflow

Inspired by the ReAct paradigm [9], **AlphaQuanter** interleaves reasoning traces with discrete *actions*, as illustrated in Figure 1. The workflow begins with an initial *Plan* generation, followed by an iterative loop with three stages: (i) identify an information gap and Acquire new evidence via a tool call; (ii) Reason over the acquired evidence to update beliefs, and (iii) Act by either continuing the loop to gather more information or committing to a trading decision. This design enforces stepwise hypothesis testing while keeping evidence collection tightly coupled to reasoning. The full prompt design is provided in Appendix B.1.

4.2 Reward Formulation

Outcome Score To train a robust, forward-looking agent under market noise, we encourage actions only on high-conviction signals, correctly classifying market states as strongly bullish, bearish, or neutral, while ignoring noise. We therefore smooth future returns by blending multiple horizons, akin to label smoothing [17, 18]. Specifically, we define the exponentially weighted forward return r_t to filter short-lived fluctuations and emphasize the medium-term trajectory:

$$r_t = \sum_{h=1}^{H} \omega_h \cdot \left(\frac{p_{t+h+1}}{p_{t+1}} - 1 \right),$$

where p_t is the asset price on day t, H is the maximum horizon, $\omega_h = \eta^h \big/ \sum_{i=1}^{H} \eta^i$ is the normalized exponential weight, and $\eta \in (0, 1)$ is the decay factor. Thresholding r_t at θ yields the market regime, and we assign discrete rewards by action as specified in Table 1.

We adopt an asymmetric penalty scheme to provide a more informative learning signal: taking the opposite side of a strong trend (reward -1.0) is penalized more than failing to act on an opportunity (reward -0.75), nudging the policy toward risk-aware behavior consistent with professional trading practice.

Table 1: Discrete reward structure for $\mathcal{R}_{\text{result}}$.

Future Market State	a_t = BUY	a_t = SELL	a_t = HOLD
Highly Bullish ($r_t > \theta$)	**+1.0**	-1.0	-0.75
Highly Bearish ($r_t < -\theta$)	-1.0	**+1.0**	-0.75
Sideways ($\lvert r_t \rvert \le \theta$)	-0.5	-0.5	**+1.0**

Process Score The process score comprises a format score $\mathcal{R}_{\text{format}}$ and a tool score $\mathcal{R}_{\text{tool}}$. The format score regulates the length of the reasoning trace: given a target token interval $[\min_{\text{token}}, \max_{\text{token}}]$, outputs outside this interval incur penalties, encouraging sufficiency without verbosity. The tool-use score governs acquisition efficiency by penalizing a total number of tool calls outside $[\min_{\text{tool}}, \max_{\text{tool}}]$. It further discourages the degenerate *collect-then-conclude* pattern, acquiring all data in a single round and immediately producing a final answer, which can cause training to collapse. In addition, malformed tool calls that violate the function signature (e.g., missing or incorrect arguments) incur additional penalties.

In conclusion, we define the total reward $\mathcal{R} = \alpha\mathcal{R}_{\text{result}} + \mathcal{R}_{\text{format}} + \mathcal{R}_{\text{tool}}$, where the hyperparameter α places a greater emphasis on the outcome score, reflecting its primary importance.

5 Evaluation

5.1 Backtesting Protocol

While the policy is optimized for day-to-day decisions, its ultimate value is determined by the strategy's risk-adjusted performance over extended horizons [19]. To evaluate this, we conduct systematic backtesting simulations in which the daily-trained policy π is applied sequentially across a historical period, generating a series of trades from which we measure the resulting portfolio performance.

5.2 Portfolio State and Transition Dynamics

Table 2: Backtesting Simulation Parameters

Symbol	Description
p_i	Closing price of the asset on day i.
h_i	Number of shares held at the end of day i.
c_i	Cash balance at the end of day i.
V_i	Total portfolio value at the end of day i. ($V_i = c_i + h_i \cdot p_i$)
λ	Transaction fee rate for BUY/SELL orders.
κ	Capital utilization ratio for BUY orders (slippage buffer).

Table 3: State transition rules for different actions

Action	h_{t+1}	c_{t+1}
BUY	$h_t + \left\lfloor \frac{\kappa\, c_t}{p_{t+1}} \right\rfloor$	$c_t - (1+\lambda) \left\lfloor \frac{\kappa\, c_t}{p_{t+1}} \right\rfloor p_{t+1}$
SELL	0	$c_t + (1-\lambda)\, h_t\, p_{t+1}$
HOLD	h_t	c_t

We define the core variables of our portfolio in Table 2. The transition from the portfolio state (h_t, c_t) to the new state (h_{t+1}, c_{t+1}) is determined by the action a_t, as summarized in Table 3.

5.3 Evaluation Metrics

Following prior work [7, 20, 21], we employ three widely used portfolio-level metrics as follows:

Annualized Rate of Return (ARR) measures profitability by annualizing total return:

$$\text{ARR} = (V_T / V_0)^{252/T} - 1$$

where V_0 and V_T are initial and final portfolio values, T is trading days, 252 is annual trading days.

Sharpe Ratio (SR) measures risk-adjusted performance:

$$\text{SR} = \bar{r} / \sigma_r$$

where $r_t = \frac{V_t - V_{t-1}}{V_{t-1}}$, $\bar{r} = \frac{1}{T}\sum_{t=1}^{T} r_t$, and $\sigma_r = \sqrt{\frac{1}{T-1}\sum_{t=1}^{T}(r_t - \bar{r})^2}$. Higher SR indicates better risk-adjusted returns.

Maximum Drawdown (MDD) measures the largest peak-to-trough decline:

$$\text{MDD} = \max_{1 \le t \le T} \left(\frac{\max_{1 \le s \le t} V_s - V_t}{\max_{1 \le s \le t} V_s} \right)$$

Lower MDD reflects better downside risk control.

6 Experimental Setup

6.1 Dataset and Simulation Period

We focus on five large-cap, event-driven stocks: Alphabet Inc. (**GOOGL**); Microsoft Corporation (**MSFT**); Meta Platforms, Inc. (**META**); NVIDIA Corporation (**NVDA**); and Tesla, Inc. (**TSLA**). These firms operate in information-dense markets and generate rich, rapidly evolving signals that stress an agent's capacity for iterative information gathering and analysis. Their volatility and frequent news shocks also create settings where dynamic planning outperforms fixed policies. To avoid look-ahead bias and mirror a realistic research-to-deployment workflow, we partition the data chronologically into non-overlapping training, validation, and test sets, as shown in Table 4. Crucially, we insert an approximately 30-trading-day gap between successive sets to eliminate leakage from overlapping feature windows and to strengthen the test of out-of-sample generalization.

6.2 Information Sources

The tool integrates four primary data categories summarized below. More details are provided in Appendix A.

Table 4: Dataset splits and trading-day counts.

Set	Start Date	End Date	#Trading Days
Training	2022-09-01	2024-03-30	395
Validation	2024-05-15	2024-11-14	128
Test	2025-01-01	2025-06-30	122

Market Data Daily price data (e.g., Open, High, Low, Close, Volume) for each stock, along with a curated set of technical indicators grouped by function: trend (e.g., SMA, EMA), momentum (e.g., RSI, STOCH), volatility (e.g., BBANDS), and volume (e.g., OBV). These support technical analysis of market dynamics.

Fundamental Data Financial data from corporate filings, including income statements, balance sheets, cash flow statements, insider trading activity, dividend history, and earnings estimates. These support the assessment of company intrinsic value.

Sentiment Data Textual signals from financial news and social media platforms (e.g., Reddit) to quantify market sentiment and investor psychology. These signals capture short-horizon sentiment and narrative shifts.

Macroeconomic Indicators Series characterizing the broader economy, including inflation (e.g., CPI), interest rates (e.g., federal funds rate), and industry activity (e.g., commodity prices). These indicators provide the macroeconomic context for asset pricing.

6.3 Implementation

Baselines We compare with four categories of baselines: (1) a passive *buy and hold* strategy; (2) classic rule-based strategies, including MACD and ZMR [7]; (3) TradingAgent [6], a state-of-the-art multi-agent LLM framework that generates trading decisions via debate-style reasoning without an explicit RL training phase; and (4) an ablated variant of AlphaQuanter that operates via zero-shot reasoning. For the LLM-based baselines, we use multiple backbone models, including Qwen2.5-3B-Instruct, Qwen2.5-7B-Instruct [22], Qwen3-30B-A3B-Instruct [23], DeepSeek-V3.1 [24], Kimi-K2 [25], GPT-4o-mini, and GPT-4o [26].

Training Details We train the AlphaQuanter agents using Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct [22] as backbones, with the verl framework [27], optimizing the policy with the GRPO algorithm [28]. At inference time, we use deterministic decoding (temperature = 0). For each configuration, we report the mean performance over three independent runs (distinct random seeds) to mitigate variance. All experiments are conducted on NVIDIA A100 GPUs (80GB). A complete list of hyperparameters and training settings is provided in Appendix B.

7 Results and Analysis

7.1 Overall Performance Comparison

To systematically evaluate current LLM-based trading paradigms and demonstrate the effectiveness of our approach, we address three research questions. Detailed metrics for all methods are summarized in Table 5.

RQ1: Single or multi-agent, which is better? We compare single-agent and multi-agent frameworks across multiple LLM backbones. The results show that, except for GPT-4o, the single-agent framework consistently outperforms the multi-agent framework on key metrics, particularly ARR and SR. This supports our hypothesis that, for smaller-scale models, multi-agent "debate" can inject noise or amplify hallucinations rather than yield complementary insights, ultimately degrading performance. These results provide clear justification for adopting a single-agent architecture in our approach.

RQ2: Is prompt-based reasoning sufficient for trading decisions? We compare the strongest prompt-based baselines against the simple *buy-and-hold* strategy. On average, all backbones except GPT-4o fail to beat the market method. We attribute the underperformance to difficulty learning actionable decision boundaries. Although the models can infer bullish or bearish sentiment, the prompt-based baseline does not reliably calibrate the decision threshold at which a signal should trigger `BUY` rather than `HOLD`. This exposes a fundamental limitation of current small-scale LLMs for trading and indicates that prompt-only reasoning is insufficient; agents should be explicitly trained to map high-dimensional market states to optimal trading actions.

RQ3: How effective is the AlphaQuanter? We compare the fully trained **AlphaQuanter** against all baselines. Both the 3B and 7B variants significantly outperform the strongest baseline, with absolute ARR gains of 6.54% and 18.45%, respectively. Moreover, the 7B model is notably consistent, outperforming all baselines on three of the five stocks. These results show that end-to-end RL training enables small LLMs to learn robust trading policies, including proactive tool use and information

Table 5: Backtesting performance comparison of different methods over a 122-day backtesting period. For the ARR of each stock and the overall average, we mark the highest value in **bold red** and the second-highest in **bold orange**.

Model	GOOGL	META	MSFT	NVDA	TSLA	Average		
	ARR (↑)	*ARR* (↑)	*ARR* (↑)	*ARR* (↑)	*ARR* (↑)	*ARR* (↑)	*SR* (↑)	*MDD* (↓)
◆ ***Market***								
Buy and Hold	-14.49%	45.64%	36.80%	25.47%	-28.91%	12.90%	0.57	31.13%
◆ ***Rule-Based***								
MACD	-3.17%	46.82%	-9.58%	-12.89%	22.77%	8.79%	0.44	21.24%
ZMR	-2.26%	-0.98%	8.53%	**35.01%**	16.74%	11.41%	0.46	20.86%
◆ ***Multi-Agent***								
Qwen2.5-3B-Instruct	1.73%	36.25%	40.89%	-3.28%	-76.98%	-0.28%	-0.13	20.95%
Qwen2.5-7B-Instruct	**9.33%**	28.98%	-4.50%	-17.22%	-9.11%	1.50%	-0.08	6.43%
Qwen3-30B-A3B-Instruct	-18.09%	1.36%	9.84%	10.22%	-16.51%	-2.64%	0.06	22.20%
DeepSeek-V3.1	-12.43%	-9.48%	14.13%	-24.02%	0.00%	-6.36%	-0.26	12.49%
Kimi-K2	-23.40%	-9.52%	12.60%	-8.33%	8.88%	-3.95%	-0.11	26.62%
GPT-4o-mini	-18.08%	0.73%	16.27%	-5.38%	5.20%	-0.25%	-0.06	18.28%
GPT-4o	-14.95%	29.69%	38.62%	-7.83%	**36.92%**	16.49%	0.50	21.82%
◆ ***Single-Agent***								
Qwen2.5-3B-Instruct	**3.06%**	23.08%	5.10%	-7.43%	-32.21%	-1.68%	0.08	25.99%
Qwen2.5-7B-Instruct	-22.42%	35.50%	17.55%	1.47%	-9.63%	4.49%	0.16	28.96%
Qwen3-30B-A3B-Instruct	-26.33%	32.86%	37.45%	29.61%	-46.41%	5.44%	0.12	30.08%
DeepSeek-V3.1	-25.15%	32.49%	25.45%	10.30%	-1.21%	8.38%	0.24	30.70%
Kimi-K2	-40.48%	25.83%	-3.39%	-3.27%	13.05%	-1.65%	0.15	25.30%
GPT-4o-mini	-24.02%	44.42%	**43.42%**	13.61%	-43.71%	6.74%	0.25	26.78%
GPT-4o	-9.01%	**57.18%**	19.39%	17.60%	-38.04%	9.42%	0.25	28.27%
◆ ***AlphaQuanter (Ours)***								
AlphaQuanter-3B	-14.68%	**56.15%**	9.82%	30.55%	33.33%	**23.03%**	0.43	25.16%
AlphaQuanter-7B	-2.52%	41.91%	**47.23%**	**45.41%**	**42.67%**	**34.94%**	**0.65**	24.93%

seeking, that even surpass powerful zero-shot setting such as GPT-4o. In short, the evidence indicates that specialized training paradigms may be more critical than model scale for achieving state-of-the-art performance in automated trading systems.

7.2 Training Dynamics and Validation Performance

To capture the training process comprehensively, we track (1) *training dynamics* (Figure 2), reflecting reward signals and behavioral patterns during learning; and (2) *validation performance* (Figure 3), evaluating trading performance on unseen data. This joint analysis reveals how the agents' evolving behaviors translate into practical outcomes.

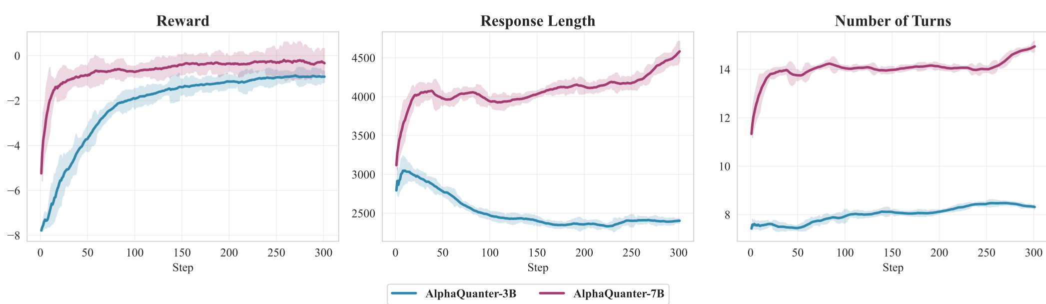

Figure 2: Comparison of training dynamics for the AlphaQuanter-3B and -7B models.

Training Dynamics The upward reward trajectory demonstrates effective learning from market interactions. However, the remaining metrics reveal markedly divergent learning trajectories between the two models. Both exhibit an initial volatile exploration phase, but their subsequent paths differ substantially. The 3B model transitions into a simplistic exploitation phase characterized by fewer tool calls and decreasing response length, suggesting premature convergence to a less robust policy. In contrast, the 7B model achieves stable exploitation around step 200 and subsequently enters a policy refinement phase, evidenced by increased response length and dialogue turns. This pattern indicates that the larger model explores more sophisticated reasoning chains and information-seeking strategies to extract marginal performance gains.

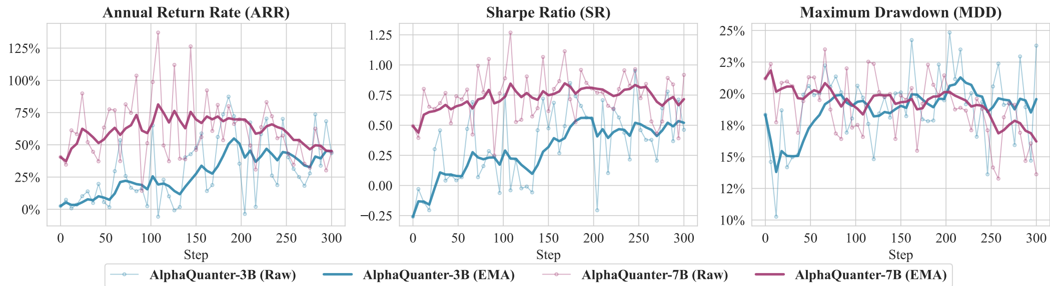

Figure 3: Comparison of key backtesting metrics for the AlphaQuanter-3B and -7B models on the validation set.

Validation Performance The validation metrics confirm the successful generalization of the learned policies to unseen data. For both 3B and 7B, ARR and SR exhibit clear upward trends that closely mirror the training reward curves. Notably, the 7B model shows a downward trend in MDD, indicating it has learned not only to maximize returns but also to effectively manage downside risk. Conversely, the 3B model's MDD oscillates with an upward bias, revealing its failure to internalize risk-aware trading behavior despite improving returns.

7.3 Tool Usage Patterns

Policy Evolution To fully understand how AlphaQuanter achieves its result and validate the interpretability of its strategies, we analyze the agent's underlying decision-making process. The heatmaps in Figure 4 show that tool usage for both the 3B and 7B models is dynamic rather than static, evolving over the course of training. This confirms that the agents actively learn and refine their information-seeking policies instead of relying on a fixed routine.

Divergent Strategies: 3B vs. 7B The two models exhibit divergent learned strategies. The 3B model exhibits a diffuse, low-contrast usage pattern across tools, suggesting limited ability to distinguish informative from uninformative signals. In contrast, the 7B model develops a concentrated, high-contrast pattern, consistent with a selective, discriminative policy for filtering and prioritizing information.

Expert-like Heuristic Closer examination of the 7B model's learned policy reveals a sophisticated, expert-level heuristic. It learns to rely heavily on *trend*, *momentum*, and *volume indicators* as primary signals, while treating *sentiment* and *macroeconomic context* as secondary but important inputs for decision-making. At the same time, it largely downweights low-frequency *fundamental* data, likely because such signals add limited value to the rapid decisions required by the task.

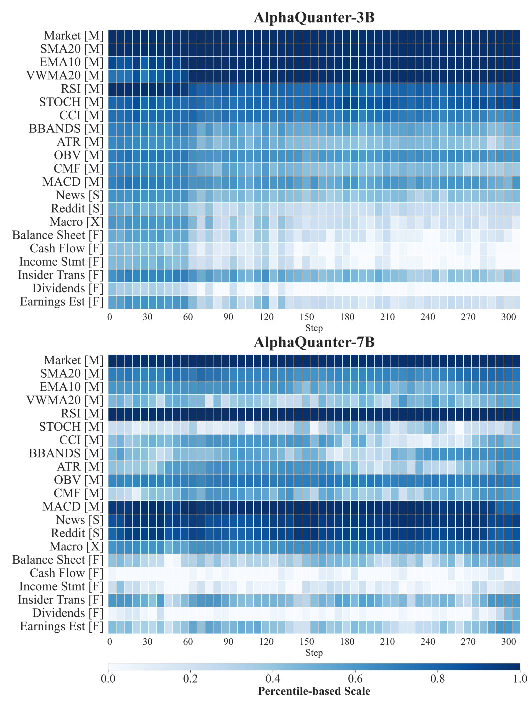

Figure 4: Evolution of the tool selection strategies for the AlphaQuanter-3B and -7B models during training. The heatmap color intensity shows the percentile-based reliance on each information at different training steps. The symbols [M], [S], [X], and [F] represent the four categories of data sources, respectively.

7.4 Ablation Studies

Table 6: Impact of reward components and the threshold θ on the performance of the AlphaQuanter-7B model.

Model	ARR (↑)	SR (↑)	MDD (↓)
AlphaQuanter-7B	**34.94%**	**0.65**	24.93%
◇ w/o $\mathcal{R}_{\text{format}}$	16.36% ($\downarrow_{53.2\%}$)	0.40	26.49%
◇ w/o $\mathcal{R}_{\text{tool}}$	19.90% ($\downarrow_{43.0\%}$)	0.49	24.08%
◇ $\theta \uparrow_{0.005}$	21.25% ($\downarrow_{39.2\%}$)	0.28	**9.18%**
◇ $\theta \downarrow_{0.005}$	20.23% ($\downarrow_{42.1\%}$)	0.43	32.67%

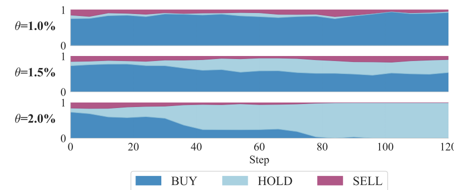

Figure 5: The effect of different decision threshold (θ) values on the agent's action distribution during training.

We conduct an ablation study to validate the contributions of our key designs, with all results shown in Table 6. First, we evaluate the effectiveness of the process score by selectively removing the format score $\mathcal{R}_{\text{format}}$ and the tool score $\mathcal{R}_{\text{tool}}$. Their removal causes the average ARR to drop by 53.2% and 43.0%, respectively, confirming their critical roles in guiding the agent toward an effective and structured decision-making process. Next, we evaluate the sensitivity of the decision threshold θ. Perturbing θ by ± 0.005 yields substantial ARR reductions of 42.1% and 39.2%. We also observe a distinct trade-off with MDD: larger θ induces more `HOLD` signals, lowering trading frequency and MDD, whereas smaller θ increases both activity and risk. As shown in Figure 5, θ is crucial for balancing exploration against exploitation; an improperly calibrated value causes the agent to converge on a single action (e.g., only `BUY` or `HOLD`), whereas our setting maintains a dynamic, adaptive policy in noisy financial environments.

8 Conclusion

In this paper, we present **AlphaQuanter**, a single-agent framework that leverages RL to directly optimize the entire decision-making process. By training the agent to learn a dynamic policy over a transparent, tool-augmented workflow, AlphaQuanter treats information acquisition as a strategic action, enabling it to adaptively reason and strategically use tools to maximize long-term profitability. Our work demonstrates that optimizing the decision-making process itself, not just a final prediction, is a crucial step toward building more robust and intelligent automated trading systems. Future work will generalize AlphaQuanter to interact with more adaptive tools in dynamic markets like real-time search, and ground its learning through long-horizon evaluation.

References

[1] David E Rumelhart, Geoffrey E Hinton, and Ronald J Williams. Learning representations by back-propagating errors. *nature*, 323(6088):533–536, 1986.

[2] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3): 273–297, 1995.

[3] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[4] John Moody and Matthew Saffell. Reinforcement learning for trading. In M. Kearns, S. Solla, and D. Cohn, editors, *Advances in Neural Information Processing Systems*, volume 11. MIT Press, 1998. URL https://proceedings.neurips.cc/paper_files/paper/1998/file/4e6cd95227cb0c280e99a195be5f6615-Paper.pdf.

[5] Zhicheng Wang, Biwei Huang, Shikui Tu, Kun Zhang, and Lei Xu. Deeptrader: a deep reinforcement learning approach for risk-return balanced portfolio management with market conditions embedding. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pages 643–650, 2021.

[6] Yijia Xiao, Edward Sun, Di Luo, and Wei Wang. Tradingagents: Multi-agents llm financial trading framework. *arXiv preprint arXiv:2412.20138*, 2024.

[7] Wentao Zhang, Lingxuan Zhao, Haochong Xia, Shuo Sun, Jiaze Sun, Molei Qin, Xinyi Li, Yuqing Zhao, Yilei Zhao, Xinyu Cai, Longtao Zheng, Xinrun Wang, and Bo An. A multimodal

foundation agent for financial trading: Tool-augmented, diversified, and generalist. In Ricardo Baeza-Yates and Francesco Bonchi, editors, *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining, KDD 2024, Barcelona, Spain, August 25-29, 2024*, pages 4314–4325. ACM, 2024. doi: 10.1145/3637528.3671801. URL https://doi.org/10.1145/3637528.3671801.

[8] Saizhuo Wang, Hang Yuan, Leon Zhou, Lionel M Ni, Heung-Yeung Shum, and Jian Guo. Alpha-gpt: Human-ai interactive alpha mining for quantitative investment. *arXiv preprint arXiv:2308.00016*, 2023.

[9] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023. URL https://openreview.net/forum?id=WE_vluYUL-X.

[10] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

[11] Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

[12] Lin Zhong. Advancements and applications of artificial intelligence in stock market prediction. 2025.

[13] Yang Liu, Qi Liu, Hongke Zhao, Zhen Pan, and Chuanren Liu. Adaptive quantitative trading: An imitative deep reinforcement learning approach. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 2128–2135, 2020.

[14] Fengchen Gu, Zhengyong Jiang, Ángel F. García-Fernández, Angelos Stefanidis, Jionglong Su, and Huakang Li. MTS: A deep reinforcement learning portfolio management framework with time-awareness and short-selling. *CoRR*, abs/2503.04143, 2025. doi: 10.48550/ARXIV.2503.04143. URL https://doi.org/10.48550/arXiv.2503.04143.

[15] Guojun Xiong, Zhiyang Deng, Keyi Wang, Yupeng Cao, Haohang Li, Yangyang Yu, Xueqing Peng, Mingquan Lin, Kaleb E Smith, Xiao-Yang Liu, et al. Flag-trader: Fusion llm-agent with gradient-based reinforcement learning for financial trading. *arXiv preprint arXiv:2502.11433*, 2025.

[16] Yijia Xiao, Edward Sun, Tong Chen, Fang Wu, Di Luo, and Wei Wang. Trading-r1: Financial trading with llm reasoning via reinforcement learning. *arXiv preprint arXiv:2509.11420*, 2025.

[17] Christian Szegedy, Vincent Vanhoucke, Sergey Ioffe, Jonathon Shlens, and Zbigniew Wojna. Rethinking the inception architecture for computer vision. In *2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016*, pages 2818–2826. IEEE Computer Society, 2016. doi: 10.1109/CVPR.2016.308. URL https://doi.org/10.1109/CVPR.2016.308.

[18] Xiao-Yang Liu, Hongyang Yang, Qian Chen, Runjia Zhang, Liuqing Yang, Bowen Xiao, and Christina Dan Wang. Finrl: A deep reinforcement learning library for automated stock trading in quantitative finance. *CoRR*, abs/2011.09607, 2020. URL https://arxiv.org/abs/2011.09607.

[19] Harry Markowitz. Portfolio selection. *The Journal of Finance*, 7(1):77–91, 1952. ISSN 00221082, 15406261. URL http://www.jstor.org/stable/2975974.

[20] Molei Qin, Shuo Sun, Wentao Zhang, Haochong Xia, Xinrun Wang, and Bo An. Earnhft: Efficient hierarchical reinforcement learning for high frequency trading. In Michael J. Wooldridge, Jennifer G. Dy, and Sriraam Natarajan, editors, *Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada*, pages 14669–14676. AAAI Press, 2024. doi: 10.1609/AAAI.V38I13.29384. URL https://doi.org/10.1609/aaai.v38i13.29384.

[21] Guojun Xiong, Zhiyang Deng, Keyi Wang, Yupeng Cao, Haohang Li, Yangyang Yu, Xueqing Peng, Mingquan Lin, Kaleb E. Smith, Xiao-Yang Liu, Jimin Huang, Sophia Ananiadou, and Qianqian Xie. FLAG-TRADER: fusion llm-agent with gradient-based reinforcement learning for financial trading. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and

Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics, ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 13921–13934. Association for Computational Linguistics, 2025. URL https://aclanthology.org/2025.findings-acl.716/.

[22] An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *CoRR*, abs/2412.15115, 2024. doi: 10.48550/ARXIV.2412.15115. URL https://doi.org/10.48550/arXiv.2412.15115.

[23] An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jian Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang, Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report. *CoRR*, abs/2505.09388, 2025. doi: 10.48550/ARXIV.2505.09388. URL https://doi.org/10.48550/arXiv.2505.09388.

[24] DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Haowei Zhang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Li, Hui Qu, J. L. Cai, Jian Liang, Jianzhong Guo, Jiaqi Ni, Jiashi Li, Jiawei Wang, Jin Chen, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, Junxiao Song, Kai Dong, Kai Hu, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Lei Xu, Leyi Xia, Liang Zhao, Litong Wang, Liyue Zhang, Meng Li, Miaojun Wang, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingming Li, Ning Tian, Panpan Huang, Peiyi Wang, Peng Zhang, Qiancheng Wang, Qihao Zhu, Qinyu Chen, Qiushi Du, R. J. Chen, R. L. Jin, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, Runxin Xu, Ruoyu Zhang, Ruyi Chen, S. S. Li, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shaoqing Wu, Shengfeng Ye, Shirong Ma, Shiyu Wang, Shuang Zhou, Shuiping Yu, Shunfeng Zhou, Shuting Pan, T. Wang, Tao Yun, Tian Pei, Tianyu Sun, W. L. Xiao, and Wangding Zeng. Deepseek-v3 technical report. *CoRR*, abs/2412.19437, 2024. doi: 10.48550/ARXIV.2412.19437. URL https://doi.org/10.48550/arXiv.2412.19437.

[25] Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, Zhuofu Chen, Jialei Cui, Hao Ding, Mengnan Dong, Angang Du, Chenzhuang Du, Dikang Du, Yulun Du, Yu Fan, Yichen Feng, Kelin Fu, Bofei Gao, Hongcheng Gao, Peizhong Gao, Tong Gao, Xinran Gu, Longyu Guan, Haiqing Guo, Jianhang Guo, Hao Hu, Xiaoru Hao, Tianhong He, Weiran He, Wenyang He, Chao Hong, Yangyang Hu, Zhenxing Hu, Weixiao Huang, Zhiqi Huang, Zihao Huang, Tao Jiang, Zhejun Jiang, Xinyi Jin, Yongsheng Kang, Guokun Lai, Cheng Li, Fang Li, Haoyang Li, Ming Li, Wentao Li, Yanhao Li, Yiwei Li, Zhaowei Li, Zheming Li, Hongzhan Lin, Xiaohan Lin, Zongyu Lin, Chengyin Liu, Chenyu Liu, Hongzhang Liu, Jingyuan Liu, Junqi Liu, Liang Liu, Shaowei Liu, T. Y. Liu, Tianwei Liu, Weizhou Liu, Yangyang Liu, Yibo Liu, Yiping Liu, Yue Liu, Zhengying Liu, Enzhe Lu, Lijun Lu, Shengling Ma, Xinyu Ma, Yingwei Ma, Shaoguang Mao, Jie Mei, Xin Men, Yibo Miao, Siyuan Pan, Yebo Peng, Ruoyu Qin, Bowen Qu, Zeyu Shang, Lidong Shi, Shengyuan Shi, Feifan Song, Jianlin Su, Zhengyuan Su, Xinjie Sun, Flood Sung, Heyi Tang, Jiawen Tao, Qifeng Teng, Chensi Wang, Dinglu Wang, Feng Wang, and Haiming Wang. Kimi K2: open agentic intelligence. *CoRR*, abs/2507.20534, 2025. doi: 10.48550/ARXIV.2507.20534. URL https://doi.org/10.48550/arXiv.2507.20534.

[26] Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Madry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, Alex Paino, Alex Renzin, Alex Tachard Passos, Alexander Kirillov, Alexi Christakis, Alexis

Conneau, Ali Kamali, Allan Jabri, Allison Moyer, Allison Tam, Amadou Crookes, Amin Tootoonchian, Ananya Kumar, Andrea Vallone, Andrej Karpathy, Andrew Braunstein, Andrew Cann, Andrew Codispoti, Andrew Galu, Andrew Kondrich, Andrew Tulloch, Andrey Mishchenko, Angela Baek, Angela Jiang, Antoine Pelisse, Antonia Woodford, Anuj Gosalia, Arka Dhar, Ashley Pantuliano, Avi Nayak, Avital Oliver, Barret Zoph, Behrooz Ghorbani, Ben Leimberger, Ben Rossen, Ben Sokolowsky, Ben Wang, Benjamin Zweig, Beth Hoover, Blake Samic, Bob McGrew, Bobby Spero, Bogo Giertler, Bowen Cheng, Brad Lightcap, Brandon Walkin, Brendan Quinn, Brian Guarraci, Brian Hsu, Bright Kellogg, Brydon Eastman, Camillo Lugaresi, Carroll L. Wainwright, Cary Bassin, Cary Hudson, Casey Chu, Chad Nelson, Chak Li, Chan Jun Shern, Channing Conger, Charlotte Barette, Chelsea Voss, Chen Ding, Cheng Lu, Chong Zhang, Chris Beaumont, Chris Hallacy, Chris Koch, Christian Gibson, Christina Kim, Christine Choi, Christine McLeavey, Christopher Hesse, Claudia Fischer, Clemens Winter, Coley Czarnecki, Colin Jarvis, Colin Wei, Constantin Koumouzelis, and Dane Sherburn. Gpt-4o system card. *CoRR*, abs/2410.21276, 2024. doi: 10.48550/ARXIV.2410.21276. URL https://doi.org/10.48550/arXiv.2410.21276.

[27] Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient RLHF framework. In *Proceedings of the Twentieth European Conference on Computer Systems, EuroSys 2025, Rotterdam, The Netherlands, 30 March 2025 - 3 April 2025*, pages 1279–1297. ACM, 2025. doi: 10.1145/3689031.3696075. URL https://doi.org/10.1145/3689031.3696075.

[28] Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

A Detailed Information Sources

A.1 Market Data

Market data consists of two tiers: raw price/volume, and a curated set of popular technical indicators. This selection is carefully designed to offer the agent a comprehensive and non-redundant toolkit for analyzing market dynamics. These data are extracted via API from Yahoo Finance[1] and Alpha Vantage[2].

A.1.1 Price and Volume Data

This data consists of the daily Open, High, Low, and Close (OHLC) prices, the Adjusted Close price, and Volume for each stock, which represents the fundamental record of an asset's trading activity for a given day.

A.1.2 Technical Indicators

To facilitate a deeper analysis of the raw price data, we allow the agent to query a set of the most widely used technical indicators, which we group by their analytical function:

Trend Indicators These help identify the direction and strength of a price trend.

- **SMA(20)**: A 20-day Simple Moving Average of the price.
- **EMA(10)**: A 10-day Exponential Moving Average, giving more weight to recent prices.
- **VWMA(20)**: A 20-day Volume-Weighted Moving Average, emphasizing periods with higher trading volume.

Momentum Indicators These measure the speed and change of price movements to identify overbought or oversold conditions.

- **RSI(14)**: The 14-day Relative Strength Index.
- **STOCH(14, 3, 3)**: The Stochastic Oscillator with the parameters defining its calculation: 14 sets the look-back period for the high-low price range, the first 3 is the smoothing period for the main oscillator line (%K), and the second 3 is the moving average period for its signal line (%D).
- **CCI(21)**: The 21-day Commodity Channel Index.

Volatility Indicators These quantify the magnitude of price fluctuations.

- **BBANDS(20, 2)**: Bollinger Bands. The parameter 20 sets the period for the SMA that forms the middle band. The 2 specifies that the upper and lower bands are plotted at two standard deviations above and below this middle band, respectively.
- **ATR(14)**: The 14-day Average True Range, a measure of market volatility.

Volume Indicators These use trading volume to confirm trends or signal potential reversals.

- **OBV**: On Balance Volume, which relates price changes to volume.
- **CMF**: The Chaikin Money Flow, which measures money flow volume over a period.

Hybrid Indicator

- **MACD(12, 26, 9)**: The Moving Average Convergence Divergence, calculated by subtracting the 26-period EMA from the 12-period EMA. The 9 refers to a 9-period EMA of the MACD line itself, which serves as a "signal line" to generate trading triggers.

A.2 Fundamental Data

To ground the agent's reasoning in a company's intrinsic financial health and valuation, we integrated several categories of fundamental data extracted via Alpha Vantage API. These sources provide a

[1] https://developer.yahoo.com/api/

[2] https://www.alphavantage.co/documentation/

holistic view, covering core financial statements, forward-looking analyst expectations, and significant corporate events. The specific data types are detailed below:

- **Earnings Estimates**: Forward-looking analyst projections, including annual and quarterly estimates for Earnings Per Share (EPS) and revenue. This dataset also provides metadata such as the number of contributing analysts and their revision histories.
- **Income Statement**: Annual and quarterly income statements detailing a company's revenues, expenses, and profitability.
- **Balance Sheet**: Annual and quarterly balance sheets providing a snapshot of a company's assets, liabilities, and shareholders' equity.
- **Cash Flow**: Annual and quarterly cash flow statements that report the flow of cash from operating, investing, and financing activities, normalized to standard accounting principles.
- **Insider Transactions**: Data on historical and recent transactions of company stock executed by key stakeholders, such as executives and board members, which can serve as a signal of internal sentiment.
- **Dividends**: A record of historical dividend payments and future declared distributions, offering insight into a company's policy on returning capital to shareholders.

A.3 Sentiment Data

We incorporated two sources of sentiment data: news and Reddit.

- For news data, we source headlines, summaries, and associated sentiment scores for each stock from the Alpha Vantage API.
- For Reddit data, we retrieve the most relevant submissions from a publicly available Reddit data dump[3], focusing on content from the 11 most popular stock-trading subreddits (e.g., `wallstreetbets`, `stocks`, `Daytrading`, etc.). To manage the input context length, the content of each original post was then summarized using the Qwen3-30B-A3B-Instruct model [23].

A.4 Macroeconomic Indicators

We integrate a set of key macroeconomic indicators extracted from the Alpha Vantage API. These indicators provide context on monetary policy, inflation, and the health of the real economy. The specific data sources are as follows:

- **Treasury Yield**: Data on the yields of U.S. Treasury securities across maturities, considered as a benchmark for risk-free interest rates and future economic growth expectations.
- **Federal Funds Rate**: The target interest rate set by the U.S. Federal Reserve, provided every month. This is a primary driver of monetary policy and affects borrowing costs throughout the economy.
- **Consumer Price Index (CPI)**: Monthly data reflecting the average change in prices paid by consumers for a basket of goods and services, serving as a primary measure of inflation.
- **WTI Crude Oil Price**: The spot price of West Texas Intermediate crude oil. It reflects global energy prices, supply-demand dynamics, and inflationary pressures.
- **Copper Price**: The spot price of copper, a critical industrial metal often considered a leading indicator of global economic health and manufacturing activity.

B Implementation Details

B.1 Prompt Design

We design the prompt to promote flexible, evidence-driven exploration rather than predetermined outputs. To achieve this, we first provide a clear task description, the specific target stock and date, and available tools, with constraints on the maximum number of tool calls to ensure efficiency. Then, we enforce the designed workflow by instructing the agent to form and test hypotheses, call only one tool at a time, and clearly show its thinking process within a structured format before each action. The complete prompt is provided in the Figure 6.

[3] https://academictorrents.com/details/ba051999301b109eab37d16f027b3f49ade2de13

```
You are a professional trading strategy analyst. Your goal is to generate a
well-reasoned final trade decision (BUY/SELL/HOLD) for a given stock and date
through systematic, evidence-based exploration using all available tools. At most
8 tool calls.

You have access to the following tools - use them intentionally and iteratively to
test hypotheses and deepen your analysis:

- [MUST] get_market_data (historical OHLCV)
- [MUST] get_stock_indicators (trend indicators(SMA20, EMA10, VWMA20), momentum
(RSI, STOCH, CCI), volatility (BBANDS, ATR), and volume-based (OBV, CMF), and
hybrid(MACD))
- [OPTIONAL] get_news_data
- [OPTIONAL] get_reddit_data
- [OPTIONAL] get_macro_indicators
- [OPTIONAL] get_balance_sheet
- [OPTIONAL] get_cashflow
- [OPTIONAL] get_income_statements
- [OPTIONAL] get_insider_transactions
- [OPTIONAL] get_dividends
- [OPTIONAL] get_earnings_estimate

GUIDELINES:
## Think Like an Analyst, Not a Script.
Approach the problem creatively. There is no single fixed workflow. Use your
reasoning to form hypotheses, then leverage tools flexibly to explore, validate,
or refute your ideas. Be curious and iterative.

## Start with a High-Level Hypothesis.
Begin by outlining your initial perspective and what you aim to investigate. This
isn't a rigid plan-it's a starting point. You're encouraged to adapt as new
evidence emerges.

## Plan, Execute, Then Analyze in the format: <think> ... </think>
- First, Briefly Plan: Before calling any tool, briefly state your current
hypothesis or what you aim to learn with the next step.
- Then, Call One Tool: Execute only one tool call per step. You must wait for and
receive the result before proceeding.
- Finally, Analyze and Adapt: Interpret the result. Does it confirm your hypothesis?
Does it reveal something new? Use this insight to refine your next step.

## One Step at a Time.
You are strictly permitted to make only one tool call at a time. The subsequent
analysis and planning must be based on the returned result before any further
tool is called. This ensures a deliberate and evidence-driven investigative process.

## Conclude with a Decision.
After synthesizing all evidence, provide a clear and justified trade recommendation
in the format: <answer>BUY | SELL | HOLD</answer>

- Current date: {date}
- Target stock ticker: {stock}
```

Figure 6: Full prompt for the AlphaQuanter agent.

B.2 Hyperparameters for RL Training

We train AlphaQuanter using verl [27]. In Table 7, we list the important parameter settings for the verl framework as well as the hyperparameters referenced in this paper.

Table 7: Hyperparameters for training AlphaQuanter.

Key	Value
algorithm.use_kl_in_reward	false
actor_rollout_ref.actor.clip_ratio_low	0.1
actor_rollout_ref.actor.clip_ratio_high	0.1
actor_rollout_ref.actor.clip_ratio_c	3
actor_rollout_ref.actor.entropy_coeff	0
actor_rollout_ref.actor.kl_loss_coef	0.05
actor_rollout_ref.actor.optim.lr	1e-6
actor_rollout_ref.actor.use_kl_loss	true
actor_rollout_ref.rollout.multi_turn.max_user_turns	32
actor_rollout_ref.rollout.multi_turn.max_assistant_turns	32
actor_rollout_ref.rollout.n	16
algorithm.kl_ctrl.kl_coef	0.0
data.max_prompt_length	3072
data.max_response_length	16384
data.train_batch_size	32
H	7
λ	0.001
κ	0.9
θ	0.015
α	5
$\min_{\text{token}}$	200
$\max_{\text{token}}$	600
$\min_{\text{tool}}$	4
$\max_{\text{tool}}$	8

B.3 Detail of Baseline

For the buy-and-hold strategy, we implement it by having the agent generate a BUY signal on every trading day. For rule-based strategy, we implement two standard baselines. The first is Moving Average Convergence Divergence (MACD), a trend-following strategy that uses indicator crossovers to generate trading signals; we employ the standard (12, 26, 9) parameterization for the fast, slow, and signal periods. The second baseline is Z-score Mean Reversion (ZMR), which assumes price reversion to a historical mean. We enters a trade when the Z-score (calculated over a 20-period lookback) exceeds a threshold of 1.0 and exits upon reversion to the mean (Z-score = 0). In addition to the classic quantitative strategies, we establish two LLM-based baselines for comparison. For the multi-agent baseline, we adapt the framework from Xiao et al. [6]. In our implementation, we replace the original data sources with our four designated categories of financial data, while retaining the prompts and agent architecture as specified in the original paper. For the single-agent baseline, we design a configuration that utilizes our custom prompt structure, as shown in Figure 6, in conjunction with the same four data categories. This baseline serves to isolate the performance of a single agent with full informational access but without the RL-optimized workflow of AlphaQuanter.

C Detailed Result Analysis

C.1 Full Results of Main Table

Table 8 and Table 9 present the complete backtesting results, providing a detailed breakdown of the ARR, SR, and MDD for each individual stock summarized in Table 5. Our asset-specific analysis reveals several key findings. For GOOGL, most models struggle to generate positive returns, although a few baseline methods achieve marginal gains. For META, the majority of strategies are profitable. Notably, the single-agent version of GPT-4o achieved the highest ARR, a result matched by AlphaQuanter-3B, which does so with a superior risk profile, evidenced by a higher SR and a lower MDD. On MSFT, AlphaQuanter-7B delivers the highest ARR, concurrently achieving a strong SR and a relatively low MDD. For NVDA, the results are mixed, with returns split between positive and negative. We observe that multi-agent methods are more prone to negative returns, whereas single-agent approaches more frequently yield positive returns with SRs greater than zero, although with high MDD. Here, AlphaQuanter-7B again secure the highest ARR, with its SR and MDD being comparable to the market baseline. For TSLA, the performance is similarly divided. It is particularly noteworthy that DeepSeek-V3.1 consistently outputs a HOLD signal, resulting in zero values for all metrics. This behavior empirically validates our earlier assertion that prompting-based models struggle to differentiate between BUY and HOLD signals under uncertainty. AlphaQuanter-7B once again achieves the highest ARR with satisfactory SR and MDD.

Table 8: Performance comparison of different methods over a 122-day backtesting period (1/2): detailed results for [GOOGL, META, MSFT].

Model	GOOGL			META			MSFT		
	ARR (↑)	SR (↑)	MDD (↓)	ARR (↑)	SR (↑)	MDD (↓)	ARR (↑)	SR (↑)	MDD (↓)
♦ ***Market***									
Buy and Hold	-14.49%	-0.35	27.35%	45.64%	1.25	31.59%	36.80%	1.41	18.79%
♦ ***Rule-Based***									
MACD	-3.17%	-0.04	14.14%	46.82%	2.17	12.51%	-9.58%	-0.49	19.97%
ZMR	-2.26%	0.01	18.47%	-0.98%	0.12	15.19%	8.53%	0.56	9.59%
♦ ***Multi-Agent***									
Qwen2.5-3B-Instruct	1.73%	0.1	5.52%	36.25%	0.85	15.28%	40.89%	1.06	12.23%
Qwen2.5-7B-Instruct	**9.33%**	1.38	1.40%	28.98%	0.87	6.54%	-4.50%	-1.05	2.27%
Qwen3-30B-A3B-Instruct	-18.09%	-0.46	26.36%	1.36%	0.29	16.29%	9.84%	0.42	15.88%
DeepSeek-V3.1	-12.43%	-0.66	12.01%	-9.48%	-0.25	17.18%	14.13%	0.6	10.09%
Kimi-K2	-23.40%	-1.09	17.57%	-9.52%	-0.1	16.12%	12.60%	0.51	9.11%
GPT-4o-mini	-18.08%	-0.94	18.86%	0.73%	0.04	11.11%	16.27%	0.48	18.52%
GPT-4o	-14.95%	-0.29	25.93%	29.69%	0.71	14.05%	38.62%	0.9	19.83%
♦ ***Single-Agent***									
Qwen2.5-3B-Instruct	3.06%	0.07	18.18%	23.08%	0.52	24.91%	5.10%	0.14	14.66%
Qwen2.5-7B-Instruct	-22.42%	-0.43	28.59%	35.50%	0.56	28.49%	17.55%	0.48	19.60%
Qwen3-30B-A3B-Instruct	-26.33%	-0.5	28.39%	32.86%	0.81	28.18%	37.45%	0.87	21.15%
DeepSeek-V3.1	-25.15%	-0.47	29.77%	32.49%	0.61	34.14%	25.45%	0.64	19.94%
Kimi-K2	-40.48%	-0.39	24.67%	25.83%	0.68	21.65%	-3.39%	-0.03	19.21%
GPT-4o-mini	-24.02%	-0.56	23.20%	44.42%	0.97	23.84%	43.42%	1.1	12.92%
GPT-4o	-9.01%	-0.12	19.72%	**57.18%**	0.99	25.02%	19.39%	0.53	23.04%
♦ ***AlphaQuanter (Ours)***									
AlphaQuanter-3B	-14.68%	-0.29	25.60%	56.15%	1.08	23.75%	9.82%	0.30	21.06%
AlphaQuanter-7B	-2.52%	0.05	21.37%	41.91%	0.78	25.65%	**47.23%**	1.17	14.85%

Table 9: Performance comparison of different methods over a 122-day backtesting period (2/2): detailed results for [NVDA, TSLA] and average.

Model	NVDA			TSLA			Average		
	ARR (↑)	SR (↑)	MDD (↓)	ARR (↑)	SR (↑)	MDD (↓)	ARR (↑)	SR (↑)	MDD (↓)
♦ ***Market***									
Buy and Hold	25.47%	0.74	33.83%	-28.91%	-0.2	44.10%	12.90%	0.57	31.13%
♦ ***Rule-Based***									
MACD	-12.89%	-0.22	30.76%	22.77%	0.78	28.83%	8.79%	0.44	21.24%
ZMR	35.01%	1.03	16.72%	16.74%	0.59	44.33%	11.41%	0.46	20.86%
♦ ***Multi-Agent***									
Qwen2.5-3B-Instruct	-3.28%	-0.06	18.77%	-76.98%	-2.6	52.95%	-0.28%	-0.13	20.95%
Qwen2.5-7B-Instruct	-17.22%	-0.99	14.12%	-9.11%	-0.59	7.82%	1.50%	-0.08	6.43%
Qwen3-30B-A3B-Instruct	10.22%	0.31	23.78%	-16.51%	-0.25	28.71%	-2.64%	0.06	22.20%
DeepSeek-V3.1	-24.02%	-0.97	23.18%	0.00%	0.0	0.00%	-6.36%	-0.26	12.49%
Kimi-K2	-8.33%	-0.28	18.88%	8.88%	0.4	71.40%	-3.95%	-0.11	26.62%
GPT-4o-mini	-5.38%	0.01	36.61%	5.20%	0.1	6.30%	-0.25%	-0.06	18.28%
GPT-4o	-7.83%	0.03	38.74%	36.92%	1.17	10.56%	16.49%	0.50	21.82%
♦ ***Single-Agent***									
Qwen2.5-3B-Instruct	-7.43%	0.14	34.63%	-32.21%	-0.46	37.59%	-1.68%	0.08	25.99%
Qwen2.5-7B-Instruct	1.47%	0.22	40.24%	-9.63%	-0.04	27.88%	4.49%	0.16	28.96%
Qwen3-30B-A3B-Instruct	29.61%	0.51	33.48%	-46.41%	-1.08	39.22%	5.44%	0.12	30.08%
DeepSeek-V3.1	10.30%	0.31	39.81%	-1.21%	0.13	29.82%	8.38%	0.24	30.70%
Kimi-K2	-3.27%	0.11	34.92%	13.05%	0.36	26.05%	-1.65%	0.15	25.30%
GPT-4o-mini	13.61%	0.35	37.60%	-43.71%	-0.59	36.32%	6.74%	0.25	26.78%
GPT-4o	17.60%	0.39	38.53%	-38.04%	-0.54	35.06%	9.42%	0.25	28.27%
♦ ***AlphaQuanter (Ours)***									
AlphaQuanter-3B	30.55%	0.51	29.04%	33.33%	0.57	26.34%	23.03%	0.43	25.16%
AlphaQuanter-7B	**45.41%**	0.66	34.91%	**42.67%**	0.58	27.88%	**34.94%**	0.65	24.93%

C.2 Reward Decomposition Analysis

To complement the analysis in Section 7.2, Figure 7 displays the learning curves for the primary reward and its constituent components, the result, format, and tool scores, on the validation set during training. A key observation is that the 7B model consistently outperforms the 3B model across all scoring metrics. The result score exhibits a clear upward trend for both models, indicating a steady improvement in the accuracy of the agent's final actions. The rate of improvement gradually

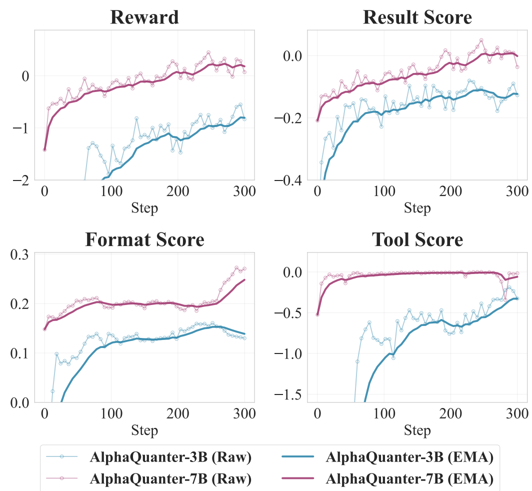

Figure 7: A comparative analysis of the training dynamics for the AlphaQuanter-3B and -7B models, illustrating the evolution of the total reward and its score components.

decelerates as the models converge. Regarding the format score, which reflects the length of the agent's reasoning trace, both models initially show an increase. However, after approximately 250 steps, their paths become different: the 7B model continues to generate more detailed reasoning, while the 3B model's reasoning length begins to decrease, leading to a decline in its format score. For the tool score, the 3B model initially performs poorly and incurs significant penalties. A case study of its rollouts reveals that in the early stages, the 3B model fails to adhere to instructions by making multiple tool calls within a single turn, which is the primary cause of its low score. This behavior is gradually rectified through further training.

C.3 Full Results of Ablation Study

Table 10: Impact of reward components and the threshold θ on the performance of the AlphaQuanter-7B model (1/2): detailed results for [GOOGL, META, MSFT].

Model	GOOGL			META			MSFT		
	ARR (↑)	SR (↑)	MDD (↓)	ARR (↑)	SR (↑)	MDD (↓)	ARR (↑)	SR (↑)	MDD (↓)
AlphaQuanter-7B	-2.52%	0.05	21.37%	41.91%	0.78	25.65%	**47.23%**	1.17	14.85%
◇ w/o $\mathcal{R}_{\text{format}}$	-6.40%	-0.09	24.86%	12.99%	0.66	25.03%	13.94%	0.51	18.93%
◇ w/o $\mathcal{R}_{\text{tool}}$	-14.22%	-0.25	25.28%	47.29%	0.85	24.23%	28.40%	0.72	19.81%
◇ $\theta \uparrow_{0.005}$	**2.83%**	0.10	4.59%	16.07%	0.27	10.91%	16.53%	0.48	2.40%
◇ $\theta \downarrow_{0.005}$	-13.05%	-0.16	28.66%	**50.82%**	0.82	34.50%	38.16%	0.87	20.01%

Table 11: Impact of reward components and the threshold θ on the performance of the AlphaQuanter-7B model (2/2): detailed results for [NVDA, TSLA] and average.

Model	NVDA			TSLA			Average		
	ARR (↑)	SR (↑)	MDD (↓)	ARR (↑)	SR (↑)	MDD (↓)	ARR (↑)	SR (↑)	MDD (↓)
AlphaQuanter-7B	**45.41%**	0.66	34.91%	**42.67%**	0.58	27.88%	**34.94%**	0.65	24.93%
◇ w/o $\mathcal{R}_{\text{format}}$	33.70%	0.49	35.55%	27.59%	0.43	28.06%	16.36%	0.40	26.49%
◇ w/o $\mathcal{R}_{\text{tool}}$	20.73%	0.43	35.24%	17.28%	0.70	15.85%	19.90%	0.49	24.08%
◇ $\theta \uparrow_{0.005}$	40.00%	0.22	20.88%	30.84%	0.32	7.14%	21.25%	0.28	9.18%
◇ $\theta \downarrow_{0.005}$	31.73%	0.53	36.50%	-6.50%	0.11	43.66%	20.23%	0.43	32.67%

Table 10 and Table 11 present the full results of our ablation study, detailing the ARR, SR, and MDD for each configuration across all five stocks. On MSFT, NVDA, and TSLA, as well as on average, all ablated or adjusted models yield inferior performance compared to the complete AlphaQuanter-7B approach. Focusing on the SR metric, AlphaQuanter-7B remains superior to all other settings on

MSFT, NVDA, and in the average case. Notably, the results demonstrate a strong inverse relationship between the decision threshold θ and MDD. Increasing the threshold consistently leads to a significant decrease in MDD across all stocks, while decreasing it results in a significant rise in MDD. This phenomenon is consistent with our analysis in Section 7.4, which attributes this behavior to changes in trading frequency.

D A Working Example

We present an example of AlphaQuanter-7B's decision-making process for **MSFT** on **May 16, 2025**, detailing its complete analysis and tool invocation trajectory.

AlphaQuanter-7B's decision-making process for MSFT on 2025-05-16

```
<think>
I will start by gathering historical market data for Microsoft (MSFT) to understand its recent price
      ↪ movements and trends. I will look back 14 days from the current date, 2025-05-16.
</think>
<tool_call>
{"name": "get_market_data", "arguments": {"symbol": "MSFT", "curr_date": "2025-05-16", "
      ↪ look_back_days": 14}}
</tool_call>user
<tool_response>
          Date     Open    High     Low    Close  Adj Close    Volume
0   2025-05-02  431.74  439.44  429.99  435.28     434.48  30757400
1   2025-05-05  432.87  439.50  432.11  436.17     435.37  20136100
2   2025-05-06  432.20  437.73  431.17  433.31     432.52  15104200
3   2025-05-07  433.84  438.12  431.11  433.35     432.56  23295300
4   2025-05-08  437.93  443.67  435.66  438.17     437.37  23491300
5   2025-05-09  440.00  440.74  435.88  438.73     437.93  15324200
6   2025-05-12  445.94  449.37  439.78  449.26     448.44  22821900
7   2025-05-13  447.78  450.67  445.36  449.14     448.32  23618800
8   2025-05-14  448.14  453.90  448.14  452.94     452.11  19902800
9   2025-05-15  450.77  456.19  450.43  453.13     453.13  21992300
10  2025-05-16  452.05  454.36  448.73  454.27     454.27  23849800
</tool_response>
assistant
<think>
From the market data, it appears that MSFT has been showing a bullish trend over the past 14 days,
      ↪ with the price closing above the previous close on most days. The volume also seems to be
      ↪ increasing, which could indicate growing interest in the stock. Next, I will analyze the
      ↪ technical indicators to get a more nuanced view of the stock's momentum and volatility.
</think>
<tool_call>
{"name": "get_stock_indicators", "arguments": {"symbol": "MSFT", "indicator": "RSI", "curr_date":
      ↪ "2025-05-16", "look_back_days": 14}}
</tool_call>user
<tool_response>
## RSI values from 2025-05-02 to 2025-05-16:

71.99-> 72.23-> 70.16-> 70.17-> 71.76-> 71.94-> 75.24-> 75.14-> 76.30-> 76.62-> 76.99

RSI: Measures momentum to flag overbought/oversold conditions. Usage: Apply 70/30 thresholds and
      ↪ watch for divergence to signal reversals.
</tool_response>
assistant
<think>
The RSI values for MSFT over the past 14 days have been consistently above 70, indicating that the
      ↪ stock is in an overbought condition. This suggests that the stock might be due for a
      ↪ correction. However, the RSI has been trending slightly lower, which could indicate that the
      ↪ selling pressure is increasing. I will now look at the Bollinger Bands to see if the stock is
      ↪  trading above the upper band, which would also suggest overbought conditions.
</think>
<tool_call>
{"name": "get_stock_indicators", "arguments": {"symbol": "MSFT", "indicator": "BBANDS", "curr_date":
      ↪ "2025-05-16", "look_back_days": 14}}
</tool_call>user
<tool_response>
## BBANDS values from 2025-05-02 to 2025-05-16:

(Middle=382.60,Upper=423.34,Lower=341.87)-> (Middle=386.41,Upper=431.68,Lower=341.15)-> (Middle
      ↪ =390.18,Upper=437.58,Lower=342.78)-> (Middle=394.11,Upper=441.87,Lower=346.35)-> (Middle
      ↪ =396.49,Upper=447.76,Lower=345.22)-> (Middle=399.36,Upper=453.11,Lower=345.61)-> (Middle
      ↪ =402.39,Upper=459.90,Lower=344.88)-> (Middle=405.45,Upper=465.82,Lower=345.08)-> (Middle
      ↪ =408.81,Upper=471.67,Lower=345.94)-> (Middle=412.92,Upper=476.08,Lower=349.75)-> (Middle
      ↪ =417.27,Upper=479.21,Lower=355.33)

Bollinger Bands: Consist of a Middle Band (typically a 20-period SMA) and Upper/Lower Bands set at
      ↪ Âś2 standard deviations from the middle. Usage: The middle band serves as a dynamic benchmark
      ↪  for price, the upper band highlights potential overbought or breakout zones, and the lower
      ↪ band signals possible oversold conditions.
</tool_response>
assistant
```

```
<think>
The Bollinger Bands for MSFT show that the stock price has been trading above the upper band for
      ↪ several days, which is a strong indication of overbought conditions. This aligns with the RSI
      ↪  readings, suggesting that the stock might be due for a correction. However, the price is
      ↪ still within the upper band, which means it's not yet at extreme overbought levels. I will
      ↪ now check the Moving Average Convergence Divergence (MACD) to see if there are any signals
      ↪ that might indicate a potential trend reversal.
</think>
<tool_call>
{"name": "get_stock_indicators", "arguments": {"symbol": "MSFT", "indicator": "MACD", "curr_date":
      ↪ "2025-05-16", "look_back_days": 14}}
</tool_call>user
<tool_response>
## MACD values from 2025-05-02 to 2025-05-16:

(MACD=7.65,Signal=0.92,Histogram=6.73)-> (MACD=10.16,Signal=2.77,Histogram=7.40)-> (MACD=11.79,Signal
      ↪ =4.57,Histogram=7.22)-> (MACD=12.94,Signal=6.24,Histogram=6.69)-> (MACD=14.07,Signal=7.81,
      ↪ Histogram=6.26)-> (MACD=14.84,Signal=9.22,Histogram=5.63)-> (MACD=16.12,Signal=10.60,
      ↪ Histogram=5.52)-> (MACD=16.92,Signal=11.86,Histogram=5.06)-> (MACD=17.67,Signal=13.02,
      ↪ Histogram=4.64)-> (MACD=18.06,Signal=14.03,Histogram=4.03)-> (MACD=18.26,Signal=14.88,
      ↪ Histogram=3.38)

MACD: Momentum indicator composed of the MACD line (difference between two EMAs), the Signal line (
      ↪ EMA of the MACD line), and the Histogram (gap between MACD and Signal). Usage: Identify trend
      ↪  changes through MACDâĂŞSignal crossovers, gauge momentum strength via Histogram size, and
      ↪ watch for divergence between MACD and price as early reversal signals.
</tool_response>
assistant
<think>
The MACD values for MSFT show a bullish trend, with the MACD line consistently above the Signal line
      ↪ and the Histogram indicating strong upward momentum. However, the recent readings of the
      ↪ Histogram suggest that the momentum might be starting to weaken. This, combined with the
      ↪ overbought conditions indicated by the RSI and Bollinger Bands, suggests that the stock might
      ↪  be due for a correction. Given the current bullish momentum, it's possible that the stock
      ↪ could see a pullback before continuing its upward trend. However, the recent weakening of the
      ↪  momentum and the overbought conditions suggest that a pullback is likely.
</think>
<tool_call>
{"name": "get_news_data", "arguments": {"symbol": "MSFT", "curr_date": "2025-05-16"}}
</tool_call>user
<tool_response>
## MSFT News, from 2025-05-14 to 2025-05-16:
Interpret the sentiment score x: values near 0 are Neutral, larger positive values indicate
      ↪ increasingly Bullish, and larger negative values indicate increasingly Bearish.
2025-05-14 [Sentiment score = 0.01, Neutral] Should You Forget Alphabet and Buy These 2 Tech Stocks
      ↪ Instead?
2025-05-14 [Sentiment score = 0.27, Somewhat-Bullish] 'Bullish For Nvidia, Tesla, Palantir, Microsoft
      ↪ , Alphabet, Amazon:' AI 'Revolution Is Coming' To The Saudi Kingdom, Says Wedbush's Dan Ives
      ↪ - Alphabet  ( NASDAQ:GOOG ) , Amazon.com  ( NASDAQ:AMZN )
2025-05-14 [Sentiment score = 0.22, Somewhat-Bullish] Correction or Not: This Artificial Intelligence
      ↪   ( AI )  Stock Is Worth Buying for the Long Haul
2025-05-14 [Sentiment score = 0.04, Neutral] The Zacks Analyst Blog Highlights C3.ai, UiPath,
      ↪ Microsoft, Alphabet and Amazon
2025-05-14 [Sentiment score = 0.13, Neutral] Alphabet Down 16% YTD: Are GOOGL Shares Buy, Sell or
      ↪ Hold on the Dip?
2025-05-14 [Sentiment score = 0.02, Neutral] Marc Benioff of Salesforce: 'You're going to have to
      ↪ throw a lot against the wall before you figure out what sticks'
2025-05-14 [Sentiment score = 0.09, Neutral] How To Trade SPY and Top Tech Stocks After a Softer
      ↪ Inflation Print - Microsoft  ( NASDAQ:MSFT )
2025-05-14 [Sentiment score = -0.22, Somewhat-Bearish] Caught Off Guard, Institutions Chase Stock
      ↪ Market Rally - SPDR S&P 500  ( ARCA:SPY )
2025-05-14 [Sentiment score = -0.03, Neutral] Nine Carahsoft Executives Recognized on the Prestigious
      ↪  2025 CRNÂő Channel Chiefs List
2025-05-15 [Sentiment score = 0.12, Neutral] Nvidia Just Rejoined Apple and Microsoft in the $3
      ↪ Trillion Club. Could It Become the First $4 Trillion Stock by 2026?

...

2025-05-16 [Sentiment score = 0.14, Neutral] Steve Ballmer's Missed Call: $1000 Investment Then Worth
      ↪  $289,000 To Apple Investors Today - Why Ex-Microsoft CEO Once Admitted 'Religious
      ↪ Transformation' Came Too Late - Microsoft  ( NASDAQ:MSFT ) , Apple  ( NASDAQ:AAPL )
2025-05-16 [Sentiment score = 0.20, Somewhat-Bullish] Satya Nadella Once Bet Big On ChatGPT-Maker
      ↪ OpenAI - Now He's Reportedly Embracing Rivals Like DeepSeek And Building Cheaper AI Models To
      ↪  Future-Proof Microsoft - Alphabet  ( NASDAQ:GOOG ) , Alphabet  ( NASDAQ:GOOGL )
2025-05-16 [Sentiment score = 0.18, Somewhat-Bullish] Why Is CoreWeave Stock Surging Today? -
      ↪ CoreWeave  ( NASDAQ:CRWV )
2025-05-16 [Sentiment score = 0.15, Neutral] Should You Buy, Sell or Hold INOD Stock After Q1
      ↪ Earnings Beat?
2025-05-16 [Sentiment score = -0.11, Neutral] Why memories are crucial for AI chatbots
2025-05-16 [Sentiment score = -0.04, Neutral] Why Alphabet Stock Ticked Higher Today
2025-05-16 [Sentiment score = 0.00, Neutral] Look Ahead - Options Expiration May Bring A Change,
      ↪ Aerospace Stocks Are On Fire From Trump Deals
2025-05-16 [Sentiment score = 0.17, Somewhat-Bullish] How To Trade SPY And Top Tech Stocks Today
2025-05-16 [Sentiment score = 0.20, Somewhat-Bullish] Prediction: This "Magnificent Seven" Stock Will
      ↪  Be the Most Valuable Company in the World in 10 Years
2025-05-16 [Sentiment score = 0.21, Somewhat-Bullish] CloudKeeper Launches 30-Day Challenge to Help
      ↪ Businesses Slash AWS Costs

</tool_response>
assistant
```

```
<think>
The news sentiment for MSFT over the past few days is mixed, with a few neutral and slightly bullish articles. However, the overall sentiment is not strongly bullish, which aligns with the technical indicators suggesting overbought conditions and potential for a pullback. The bullish news could provide some support for the stock, but the technical indicators suggest that the stock might need a correction.

Given the overbought conditions and the recent weakening of momentum, I will consider the possibility of a pullback. However, the strong support from bullish news could mean that the pullback might be short-lived. I will now check the insider transactions to see if there are any significant buying or selling activities by company insiders, which can provide insights into the management's view of the stock.
</think>
<tool_call>
{"name": "get_insider_transactions", "arguments": {"symbol": "MSFT", "curr_date": "2025-05-16", "look_back_days": 7}}
</tool_call>user
<tool_response>
## MSFT insider transactions from 2025-05-09 to 2025-05-16:
### Transaction Date: 2025-05-15, COLEMAN, AMY (EVP, Chief Human Resources Off)
Type: Common Stock
Shares: 77.894 (Disposal)

### Transaction Date: 2025-05-15, COLEMAN, AMY (EVP, Chief Human Resources Off)
Type: Common Stock
Shares: 13242.774 (Disposal)


</tool_response>
assistant
<think>
The insider transactions for MSFT show that Amy Coleman, the EVP of Human Resources, has been disposing of a significant number of shares, which could indicate a lack of confidence in the stock. This is a negative signal for the stock and could contribute to the overbought conditions. Given the technical indicators and the recent insider selling, it suggests that the stock might be due for a correction. However, the bullish news and the strong support from the market data might mean that the correction could be short-lived.

Considering the current overbought conditions, the recent insider selling, and the potential for a pullback, I will recommend a HOLD strategy. This allows for the stock to potentially correct and provides a safer entry point for a more bullish stance in the future.
</think>
<answer>HOLD</answer>
```