

Language Model Guided Reinforcement Learning in Quantitative Trading

Adam Darmanin
 University of Malta
 Msida, Malta
 adam.darmanin.03@um.edu.mt

Vince Vella
 University of Malta
 Msida, Malta
 vvell04@um.edu.mt

Abstract—Algorithmic trading requires short-term tactical decisions consistent with long-term financial objectives. Reinforcement Learning (RL) has been applied to such problems, but adoption is limited by myopic behaviour and opaque policies. Large Language Models (LLMs) offer complementary strategic reasoning and multi-modal signal interpretation when guided by well-structured prompts.

This paper proposes a hybrid framework in which LLMs generate high-level trading strategies to guide RL agents. We evaluate (i) the economic rationale of LLM-generated strategies through expert review, and (ii) the performance of LLM-guided agents against unguided RL baselines using Sharpe Ratio (SR) and Maximum Drawdown (MDD).

Empirical results indicate that LLM guidance improves both return and risk metrics relative to standard RL.

Index Terms—Large Language Models, Reinforcement Learning, Algorithmic Trading, Prompt Engineering, Agents

I. INTRODUCTION

Algorithmic trading requires short-term execution aligned with long-term financial objectives, and accounts for over 60% of U.S. equity volume, particularly in high-frequency trading (HFT) [1]. Long-horizon strategies often build on econometric models such as the Fama–French five-factor framework [2], while short-horizon approaches exploit transient inefficiencies through momentum and mean reversion [3]. The most extreme form is HFT, where firms execute thousands of transactions per second by exploiting order book imbalances [1].

Modern trading systems leverage machine learning (ML) to process structured and unstructured data streams. Growth is driven by advances in electronic infrastructure, compute power, and the proliferation of large high-resolution financial data [4].

RL formalises trading as sequential decision making [5] and has shown promise through methods such as Deep Q-Networks (DQN) and actor–critic variants [3], [6]. Yet practical adoption is hindered by sparse rewards, credit assignment issues, and opaque policies, which reduce trust in high-stakes financial settings [7].

LLMs offer complementary strengths. They can understand heterogeneous signals and generate rationale explanations [8], yet remain limited by fixed knowledge cut-offs and an inability to adapt to changing environments in real time [9]. They are also fragile to prompt design and may produce plausible but invalid outputs [10], [11]. Advances in prompt engineering and

human-in-the-loop (HITL) methods partially mitigate these risks [12].

This paper addresses these limitations by proposing a hybrid architecture that integrates LLM strategic guidance with RL execution, and sets the following objectives.

a) Objectives: (1) Design an approach for using LLMs to generate trading strategies that are economically grounded, assessed by expert review; (2) evaluate whether LLM guidance improves RL performance, measured by SR and MDD, without altering the base RL architecture.

These objectives directly inform the main contributions of the paper.

b) Contributions: (1) A structured prompting framework that produces domain-grounded strategies assessed by expert review and financial metrics; (2) a modular LLM+RL design in which the LLM contributes a single uncertainty-weighted scalar appended to the RL observation space, improving out-of-sample performance without modifying the RL algorithm.

A. Related Work

A Trading DQN (TDQN) for stock trading is introduced in [3]. The authors leverage a DDQN algorithm to mitigate overestimation bias and stabilize learning in stochastic market environments. A key aspect of their RL approach was the discretization of actions and the enforcement of capital constraints, which helped prevent infeasible or overleveraged actions.

The FinRL framework [6] introduced benchmark environments and unified APIs for financial RL research that features realistic data simulation. It includes a wide array of backtests using standard RL algorithms and focuses on two primary objectives in algorithmic trading: maximizing return (measured by cumulative return and SR) and minimizing risk (measured by MDD and return variance). The framework supports experiments in single-stock trading, multi-stock trading, and portfolio allocation.

In the survey [13], the authors identified key limitations in deep reinforcement learning (DRL), including Bellman backup instability and credit assignment failures. The authors recommend hierarchical reinforcement learning (HRL) or recurrent extensions to address the lack of long-range temporal dependencies.

For LLMs in finance, [14] showed that ChatGPT outperforms traditional sentiment lexicons on forward-looking financial news, but lacks temporal awareness and numerical reasoning capabilities.

The FINMEM framework in [12] combines structured memory with LLM-based decision modules. FINMEM’s layered memory integrates recent news, financial reports, and long-term statements to inform trade recommendations, leveraging retrieval-augmented generation (RAG). Their architecture stores experiences in a vector database, which are retrieved and ranked using a decay mechanism that emulates a human’s memory decay.

Prompting practices have been extensively surveyed in [11], which categorizes strategies into instruction-based, example-based, reasoning-based, and critique-based families. The study highlights self-refinement and constraint enforcement as key mechanisms for improving robustness. It also shows that minor variations in prompt wording can systematically influence model behavior.

In [15], Chain-of-Thought (CoT) prompting was shown to significantly enhance LLM reasoning. Self-improvement frameworks iteratively refine rationale quality, while problem decomposition and model fine-tuning help address complex tasks. Without structured prompting techniques, however, LLMs continue to struggle with planning problems.

II. METHODOLOGY

This section outlines the methodology developed to evaluate the integration of LLMs into RL agents. The proposed hybrid framework mirrors the top-down decision-making structures common in financial institutions.

Two experiments were conducted to address the research objectives. All LLM strategies were validated through historical backtesting and expert review prior to their integration with RL agents.

A. Benchmark Environment

For our benchmark, we utilized the trading system introduced in [3]. This benchmark includes a clearly defined environment, consistent state and reward functions, and extensive empirical results.

We replicated the core experimental settings, including the asset universe, data preprocessing, and evaluation metrics. Our reproduction yielded comparable statistically significant SR and MDD metrics.

We note minor discrepancies that affect financial interpretability: their cumulative returns are arithmetically summed rather than geometrically compounded, and their SR assumes a zero risk-free rate. For consistency, we preserve these conventions throughout our experiments.

B. Experiment 1: LLM Trading Strategy Generation

This experiment addressed Objective 1 by introducing two LLM agents: the **Strategist Agent** and the **Analyst Agent**. The Strategist Agent generates global trading policies using a financial dataset. The Analyst Agent processes news and

distills it into signals to inform the Strategist Agent. This experiment serves as the foundation for Experiment 2.

A strategy defines a directional action ($\text{dir}(\pi^g)$, where 1 = LONG and 0 = SHORT) and an associated confidence score (μ_{conf} , from 1 to 3). Each strategy is accompanied by an explanation and a weighted set of features. Strategies are generated on a monthly basis using time-aligned data.

1) Data and Feature Engineering: To support strategy generation, the LLM agents consumed a multi-modal dataset spanning 2012–2020, aligning with the benchmark’s dataset dates in [3]. The dataset includes traditional Open, High, Low, Close, and Volume (OHLCV) price data, which we augmented with four additional categories of financial signals: market data, fundamentals, technical analytics, and alternative data [16]. These collectively define the LLM’s context.

Market data was sourced from Interactive Brokers¹ and iVolatility², including OHLCV time series, SPX and NDX index returns, the VIX index, and Options implied volatility (IV). Fundamental data, comprising firm-level financial ratios and macroeconomic indicators (e.g., GDP, PMI, interest rates), were retrieved via SEC-API³ and the FRED API⁴. Analytics features were computed using TA-Lib⁵, applying rolling windows to extract indicators. Alternative data consisting of news headlines were collected from Alpaca⁶ and processed into explanatory factors using few-shot LLM prompting, following the LLMFactor framework from [10]. The data was aligned by timestamp.

2) LLM Model: We used OpenAI’s GPT-4o Mini for its strong performance in financial reasoning and cost-efficiency [12], [14], [15]. The model supports a 128k token context window with a 16k maximum prompt size, enabling the use of detailed prompts with embedded context memory, reasoning chains, and previous reflection results.

3) Prompt Engineering Methodology: The objective was to construct a prompt that generalized across equities and regimes while remaining interpretable. We proceeded in three stages: (i) baseline specification, (ii) incorporation of expert exemplars with feature pruning, and (iii) iterative refinement comprising two distinct processes: a *Writer-Trainer* process (feature and instruction selection) and a *Writer-Judge* process (prompt quality and rationale critique) with regret minimization.

To manage computational cost, we did not tune on full eight-year history and instead randomly sampled non-overlapping one-year intervals from the dataset per instrument, repeated five times, and used these subsets for creating candidate prompts.

a) Baseline: We began with a minimal prompt that exposed only raw OHLCV data and a small set of technical indicators: Simple Moving Averages (SMA; 20/50/200 periods), Relative Strength Index (RSI), and Moving Av-

¹<https://www.interactivebrokers.com/api>

²<https://www.ivolatility.com/data-cloud-api/>

³<https://sec-api.io/>

⁴<https://fred.stlouisfed.org/docs/api/fred/>

⁵<https://ta-lib.org/>

⁶<https://alpaca.markets/docs/api-documentation/>

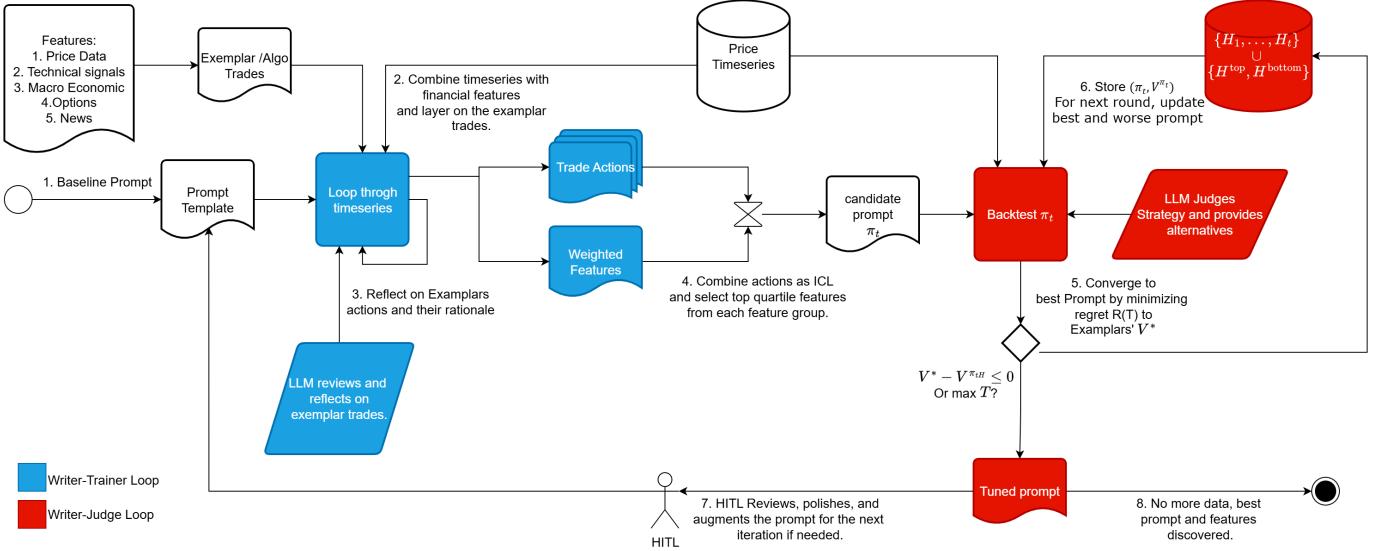


Fig. 1. Prompt Tuning Workflow.

verage Convergence Divergence (MACD). This configuration reflected common trading heuristics in both algorithmic and retail practice [17], [18] and served to set the candidate prompt and V^* target.

b) Writer-Trainer, and Writer-Judge Loops: To refine the information set, we introduced a *Writer-Trainer* process that reflected on expert trade exemplars derived from HTL feedback and a heuristic algorithm approximating these. Candidate features were selected by top quartile through ranking importance on a three-point Likert scale (low/medium/high), and rationales were clustered with only the ten unique ones being selected to become instructions for the candidate prompt.

Prompt refinement was then conducted through a heuristic regret-minimization loop, inspired by [19], with pruning and rationale discovery as the inner stage and backtesting as the outer stage. At each iteration t , a *writer* generated a candidate prompt π_t conditioned on the KB and the retained features and rationales. The candidate π_t was evaluated through backtesting to obtain its SR, denoted V^{π_t} . A *judge* then assessed the prompt for its rationale and suggested alternative instructions or feature combinations for the next iteration, and stored in the Knowledge Base (KB).

We adopted a regret heuristic to guide exploration:

$$\mathcal{R}(T) = \mathbb{E} \left[\sum_{t=1}^T (V^* - V^{\pi_t}) \mid \mathcal{H}_t \right], \quad (1)$$

where V^* denotes the best SR defined by the baseline prompt or the market's SR for the whole dataset (initialized as $V^* = \max\{V_{\text{baseline}}, 0.8\}$), V^{π_t} is the SR of the current candidate, and \mathcal{H}_t represents the KB at iteration t (features, rationales, and prior outcomes). To manage the LLM's finite context window, the loop retained only an *extremes memory* within \mathcal{H}_t , consisting of the best- and worst-performing prompts with their associated features and instructions. These extremes

biased subsequent generations toward more promising candidates.

Iterations terminated when either (i) $\mathcal{R}(T) \leq 0$ (no expected improvement) or (ii) the maximum T iterations elapsed. The final augmented baseline prompt, generalized across equities and regimes through this process, was then subjected to three additional refinements.

c) Prompt Improvement 1 – In-Context Memory (ICM): Inspired by [12], we introduced a memory buffer that stores the most recent strategies π_t observed prior to time T . Each stored strategy π_{T-1}^g is represented by its directional action, weighted features, and rationale. Within the prompt, these prior strategies are recalled in-context and reflected on, enabling the current strategy π_T^g to be conditioned on past decisions. This reflection mechanism reduces the persistence of suboptimal strategies.

d) Prompt Improvement 2 – Instruction Decomposition: To enhance reasoning, instructions and their associated drivers were decomposed [11] into six feature groups: stock, technical, fundamental, macroeconomic, options, and prior strategy reflection [9]. Each group supplied few-shot examples and domain-specific heuristics to elicit CoT, requiring the model to reason sequentially across domains and prior strategies.

e) Prompt Improvement 3 – News Factors: Unstructured news data was introduced via the analyst agent, which applied instruction-decomposition factor extraction [10]. Entities and timestamps were anonymized, disabling the LLM's memory to prevent leakage [20]. Extracted news factors were ranked and integrated alongside numerical indicators.

The final system integrates selected numerical and textual signals into a global strategy policy π^g . All prompt iterations used in Experiment 1 are summarized in Table I.

4) Parameters and Evaluation: Prompt tuning used temperature 0.7 following prior work [12], [14], with frequency penalty 1.0 and presence penalty 0.25. These values discour-

TABLE I
PROMPT VERSIONS USED IN EXPERIMENT 1

Prompt	Description
P0	Baseline prompt containing only static technical indicators and price features.
P1	Augmented P0 with selected features and instructions.
P2	P1 extended with ICM, incorporating prior strategy.
P3	P2 extended with instruction decomposition and CoT reasoning across six structured signal groups.
P4	P3 enriched with macroeconomic and firm-specific news-derived directional signal.

aged verbatim reuse from the KB or ICM while preserving exploratory diversity.

For strategy generation, temperature was set to 0 with fixed seed 49 for reproducibility. Strategies were produced on a monthly cadence (20 trading days), aligning with common guidance/rebalancing cycles and remaining tractable given LLM inference cost.

At most three refinement iterations were permitted ($T \leq 3$). Convergence was declared when the regret $\mathcal{R}(T)$ approached zero or when the SR exceeded the initial threshold $\max\{V_{\text{baseline}}, 0.8\}$. The procedure was repeated five times with discretionary HITL adjustments between runs. The iteration count balanced methodological tractability against computational cost.

All technical indicators used a 20-trading-day rolling window with standard *TA-Lib* defaults (e.g., 14-day RSI).

a) *Quantitative Metrics*: We evaluate LLM-generated strategies using three complementary metrics: risk-adjusted returns, model confidence, and model uncertainty.

The SR serves as the core risk-adjusted returns metric:

$$\text{SR} = \frac{\mathbb{E}[R_t - R_f]}{\sigma_R}. \quad (2)$$

where R_t is the portfolio return, R_f is the risk-free rate, and σ_R is the return volatility. SR also serves as a proxy for the LLM's financial reasoning [12], [14]. To ensure comparability across different periods for the daily returns, we annualize the SR to 252 trading days per year: Annualized SR = $\text{SR} \cdot \sqrt{252}$.

As a proxy for prompt quality we compute the Perplexity (PPL) [21] over the LLM-generated strategies:

$$\text{PPL} = \exp\left(-\frac{1}{N} \sum_{t=1}^N \log p(w_t | w_{<t})\right). \quad (3)$$

where $p(w_t | w_{<t})$ denotes the conditional token probability. Lower values indicate higher quality.

To complement this, we report token-level entropy H_{LLM} , approximated using top- k distributions:

$$H_{\text{LLM}} = \frac{1}{N} \sum_{t=1}^N \left(\sum_{v \in V_k} -p_t(v) \log p_t(v) - p_{\text{tail},t} \log p_{\text{tail},t} \right). \quad (4)$$

where V_k denotes the top- k token set and $p_{\text{tail},t}$ represents the unobserved probability mass [22]. In our experiments, $k = 5$.

TABLE II
EXPERT RUBRIC FOR SCORING LLM RATIONALES

Criterion	1	2	3
Rationale	Flawed	Partial	Sound
Fidelity	Unrealistic	Plausible	Professional
Safety	Ignored	Mentioned	Addressed

Lower entropy indicates greater decisiveness, whereas higher values suggest uncertainty.

Together, PPL and H_{LLM} enable a measurement of prompt quality and strategy confidence.

b) *Qualitative Evaluation*: Qualitative assessment was conducted via an Expert Review Score (ERS), a human-grounded rubric evaluating LLM-generated trading rationales along three dimensions: economic rationale, domain fidelity, and trade safety (risk awareness). Each dimension was scored on a 3-point ordinal scale {1 = poor, 2 = average, 3 = good}, based on the rubric shown in Table II.

The review process followed a similar setup to that of [23], involving ten participants: five senior finance professionals and five retail traders (or professionals in the industry who do not actively trade). Each reviewer evaluated anonymized data for three instruments over one year, including price data, fundamental and macroeconomic metrics, and firm-level news headlines.

Before reviewing the LLM rationale, expert participants made their own directional prediction (LONG/SHORT) to activate their internal domain models. They then reviewed the LLM's reasoning and scored it using the rubric. Each session concluded with a 60-minute structured discussion to elicit LLM critiques and identify exemplars to use. Surveys took approximately 15 minutes to complete. All scores were normalized to a 1–3 range.

C. Experiment 2: LLM-Guided RL

This experiment addressed Objective 2 by incorporating the LLM guidance within an RL framework.

1) *Data and Feature Engineering*: The LLM outputs from Experiment 1 were reused. The RL agent adopted the DDQN configuration of [3], with a single LLM-derived interaction term τ as innovation to the observation space. This feature consisted of:

- **Signal Direction** ($\text{dir}(\pi^g)$): The discrete directional recommendation from the LLM. Zero represents SHORT and one LONG.
- **Signal Strength** ($\text{str}(\pi^g)$): The LLM's entropy-adjusted confidence score as a Likert-3 score.

The interaction term was defined as

$$\tau = \text{dir}(\pi^g) \cdot \text{str}(\pi^g), \quad (5)$$

where $\text{dir}(\pi^g)$ was remapped from $\{0, 1\}$ to $\{-1, 1\}$ to enable the interaction.

The LLM's signal strength was derived from the normalized LLM's confidence score:

$$\mu_{\text{conf}} = \frac{\text{Likert}}{3}, \quad (6)$$

and adjusted using entropy-based certainty:

$$C = \varepsilon + (1 - \varepsilon)(1 - H), \quad (7)$$

where $H \in [0, 1]$ is the normalized entropy of the LLM output, and $\varepsilon = 0.01$ ensures numerical stability. The final strength term is:

$$\text{str}(\pi^g) = \mu_{\text{conf}} \cdot C. \quad (8)$$

This entropy-adjusted confidence follows the approach of [24], providing a soft weighting of the LLM's signal by its certainty.

The interaction term τ was selected empirically. Initial variants used direction only ($\text{str}(dir)$), followed by LLM's confidence ($\text{str}(\pi^g)$) and direction. The final form was chosen based on empirical performance and compatibility with DDQN's continuous normalized input space [3].

2) *LLM+RL Hybrid Architecture*: The baseline DDQN agent is augmented by the Strategist Agent and Analyst Agent, which produce monthly strategies for the stock's behavior. For practical reasons, outputs from the LLM were precomputed per instrument and fixed throughout training.

3) *Training and Parameters*: Hyperparameters mirror [3] and the LLM settings follow those in Experiment 1. Training was conducted over 25 runs \times 50 episodes per instrument using an NVIDIA RTX 3050, with each equity trained for 3 hours.

To ensure comparability with the benchmark [3], we replicated all baseline metrics within acceptable statistical bounds.

4) *Evaluation Metrics*: Two measures were considered:

- **SR**: Same as Experiment 1 see Eq. (2).
- **MDD**: Captures the largest observed loss from a historical peak to a subsequent trough:

$$\text{MDD} = \frac{P_{\text{peak}} - P_{\text{low}}}{P_{\text{peak}}}. \quad (9)$$

where P_{peak} is the highest portfolio value observed before the largest drop, and P_{low} is the lowest value reached before a new peak is established. Lower values indicate stronger downside protection.

These metrics together assess whether LLM-guided RL agents can adapt to different equities without changing the core architecture.

III. RESULTS AND DISCUSSION

A. Experiment 1 Results

This section presents empirical results across the baseline (P0) and four prompt versions (P1–P4) from Table I, addressing Objective 1. We evaluated their impact on SR, PPL, and H_{LLM} . From P4 onward, qualitative evaluation was incorporated through ERS, introduced once the prompt design had stabilized. All backtests were conducted over 2018–2020 to ensure comparability with the RL's OOS results. Statistical significance was assessed using two-tailed t -tests across 25 runs per ticker, with hypotheses $H_0 : \mu_{\text{P4}} = \mu_{\text{P1}}$. All runs were executed at a sampling temperature of 0.7 to capture

TABLE III
SHARPE RATIO ACROSS PROMPTS AND BENCHMARK

Ticker	P0	P1	P2	P3	P4	BM
AAPL	1.13	1.09	1.07	1.07	2.09	1.27
AMZN	0.51	0.35	0.38	0.63	0.84	0.21
GOOGL	0.34	0.26	0.52	0.52	1.12	0.19
META	0.60	-0.06	-0.28	0.30	0.77	0.63
MSFT	0.36	1.07	1.11	1.31	0.50	1.17
TSLA	0.34	0.71	0.75	0.43	0.79	0.67
Mean	0.55	0.57	0.59	0.71	1.02	0.69

TABLE IV
PERPLEXITY ACROSS PROMPTS

Ticker	P0	P1	P2	P3	P4
AAPL	1.44	1.85	1.31	1.55	1.44
AMZN	1.51	1.74	1.35	1.68	1.31
GOOGL	1.56	1.77	1.49	1.78	1.33
META	1.47	1.73	1.31	1.39	1.38
MSFT	1.43	1.83	1.44	1.49	1.24
TSLA	1.46	1.77	1.50	1.63	1.39
Mean	1.48	1.78	1.40	1.59	1.35

variance, while the reported metrics correspond to the deterministic setting (temperature 0) with fixed random seeds for reproducibility.

Tables III–V summarize the results across prompt versions relative to the benchmark (BM). Prompt 0, which relied solely on static technical features, outperformed Prompt 1 primarily because all equities exhibited upward trends during the OOS period. Prompt 1 yielded the weakest performance, with the lowest SR across most equities and the highest PPL and entropy, indicating that the LLM was unable to exploit the additional information when presented in an isolated context. Prompt 2 incorporated ICM, producing moderate gains in SR (mean 0.59) and suggesting improved confidence through reflection. Prompt 3 introduced decomposed instructions, eliciting CoT, and outperformed the benchmark with a mean SR of 0.71. Prompt 4 further included unstructured news signals and achieved the highest mean SR (1.02), lowest PPL and entropy, and showed higher confidence particularly on sentiment-sensitive tickers such as TSLA. Based on the p -values in Table VII, the improvements were statistically significant for SR and entropy, while the changes in PPL were comparatively weaker.

Expert evaluation of Prompt 4 confirmed its effectiveness. Reviewers rated the LLM's rationale highly (mean 2.7 out of 3), highlighting its ability to integrate valuation, sentiment, and analytics.

Fidelity received a slightly lower score (mean 2.65), with critiques focused on inconsistent thresholding. For instance, one reviewer noted, “Calling RSI near 40 ‘oversold’ is debatable,” requiring refinements in numerical phrasing.

Feedback varied by background: buy-side professionals emphasized transparency in feature weighting, whereas retail reviewers focused on technical and macro signals. All com-

TABLE V
ENTROPY ACROSS PROMPTS

Ticker	P0	P1	P2	P3	P4
AAPL	0.66	0.70	0.67	0.66	0.69
AMZN	0.69	0.69	0.69	0.69	0.67
GOOGL	0.67	0.67	0.67	0.70	0.66
META	0.68	0.66	0.70	0.73	0.67
MSFT	0.65	0.66	0.68	0.72	0.65
TSLA	0.67	0.68	0.70	0.74	0.65
Mean	0.67	0.68	0.69	0.71	0.67

TABLE VI
EXPERT REVIEWER SCORES FOR PROMPT 4

Dimension	ERS (1-3)
Rationale	2.70
Fidelity	2.65
Safety	2.80

mented on the lack of a neutral or hold signal, which was done to align with [3].

Overall, results validated Prompt 4’s modular design and market narrative awareness. It outperformed earlier prompts and was selected as the global policy generation prompt for the LLM–RL hybrid in Experiment 2.

The computational costs of Experiment 1 are summarized in Table VIII. The overall cost for a single run with each prompt was approximately \$36, increasing to about \$150 when the writer-judge loop was included. When accounting for additional trials and development, the cumulative cost amounted to \$345. Inference time ranged from approximately 1.5 to 2 hours per asset and prompt version.

B. Experiment 2 Results

This experiment addressed Objective 2 by comparing three agent architectures: (i) the benchmark RL-only [3], (ii) the best-performing LLM prompt from Experiment 1, and (iii) a hybrid LLM+RL agent. All agents were trained in identical environments.

To determine whether the hybrid agent outperformed the benchmark, we conducted two-sided paired *t*-tests on the SR across 25 runs for each stock. The null hypothesis H_0 assumed no difference in mean performance: $H_0: \mu_{\text{LLM+RL}} = \mu_{\text{RL-only}}$. All resulting *p*-values were below 0.05, indicating statistically significant improvements.

Results in Table IX confirm that the LLM+RL agent outperformed the RL-only baseline in four out of six assets.

AAPL and META did not show consistent individual outperformance. Fig. 2 illustrates AAPL’s trading behavior during one episode. The top panel plots price, technical indicators, and trades: hollow triangles mark RL trades; filled arrows show LLM monthly guidance. The LLM issued sparse but confident signals (strength > 0.6), often aligned with technical points of interest (e.g., MA interactions). In contrast, the RL agent frequently mistimed entries and exits.

TABLE VII
P4 vs. P1 SIGNIFICANCE OF METRIC CHANGES

Metric	<i>t</i> -test <i>p</i> -value
Entropy	2.29×10^{-4}
Perplexity	7.25×10^{-2}
Sharpe Ratio	2.3×10^{-5}

TABLE VIII
TOKEN USAGE AND COSTS

Prompt	Mean Tokens	Mean Cost(\$)	Total Tokens	Total Cost(\$)
v0	663	\$0.00020	2.0×10^6	\$1.19
v1	1,760	\$0.00043	3.5×10^6	\$5.62
v2	2,240	\$0.00051	4.5×10^6	\$6.48
v3	3,300	\$0.00067	6.6×10^6	\$8.75
v4	8,300	\$0.00150	1.6×10^7	\$21.60

From December 2018 to January 2019, the RL agent oscillated between LONG and SHORT positions with punishing results and despite receiving strong signals from the LLM. The LLM issued high-confidence guidance for a SHORT in December followed by a LONG in January, both with signal strengths exceeding 0.8. Regardless, the RL agent held a LONG position throughout the decline.

As shown in Fig. 5, the DDQN assigns lower Q-values to SHORT actions, indicating limited confidence. This follows from lower-bound constraints (used to cap leverage) that created an asymmetric return function by triggering buy-to-cover after price increases, reducing portfolio value and subsequent SHORT exposure. Also, the selected equity universe has positive historical drift, which raises average prices, with limited opportunity to capture SHORT returns. Together these features lower the expected return of a SHORT and discourage sustained SHORT positions [3].

The bottom panel confirms that the LLM maintained high confidence near key inflection points, and reduced conviction when trends have persisted (possibly awaiting a reversal from its training corpus). However, the RL agent didn’t fully exploit these signals due to the underlying RL architecture, which remained fixed for the purposes of this experiment.

Fig. 3 illustrates the evolution of the SR for AAPL throughout the training episodes. The hybrid LLM+RL agent (orange line) outperformed the baseline RL agent (blue line) in both mean Sharpe and stability, as reflected in the narrower shaded confidence intervals. The LLM’s SR is shown for reference (black dashed line).

Figs. 4 and 5 show Q-values for LONG and SHORT actions respectively, with y-axis clipped to $[-0.03, 0.03]$ to highlight late-episode convergence. Early training was noisy for both agents. The LLM+RL agent converged faster with lower variance. Although Q-value separation rarely exceeded 0.01, the hybrid showed slightly stronger directional signals. These gains emerged without modifying the DDQN or imposing reward shaping, thus isolating the effect of the LLM’s guidance. The narrow Q-range stems from the RL baseline design.

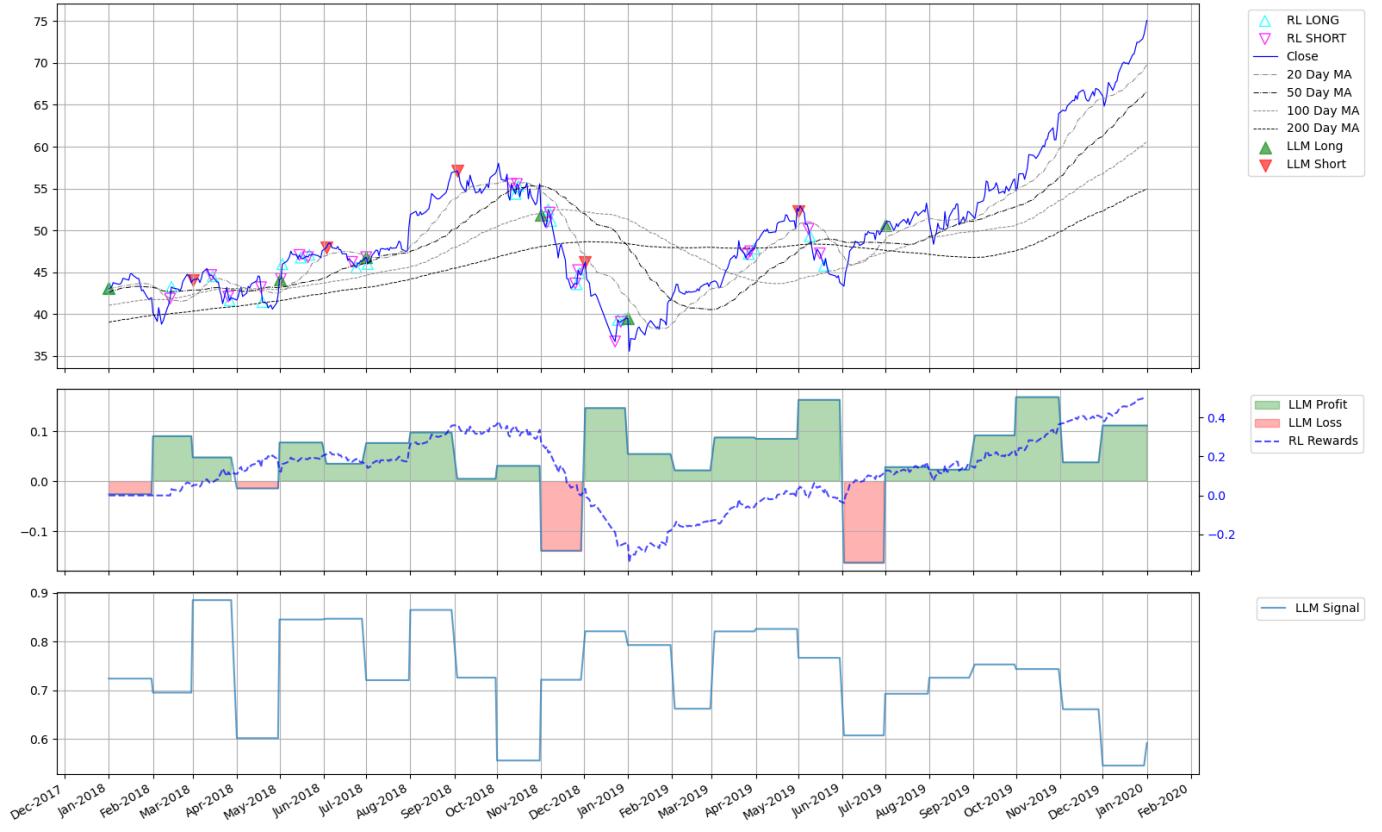


Fig. 2. AAPL Performance with LLM+RL Model.

TABLE IX
EXPERIMENT 2 RESULTS: SHARPE RATIO

Ticker	LLM+RL (σ)	RL-Only (σ)	LLM-Only
AAPL	1.70 (0.43)	1.42 (0.05)	2.09
AMZN	1.21 (0.58)	0.42 (0.23)	0.84
GOOGL	1.16 (0.17)	0.23 (0.37)	1.12
META	0.46 (0.75)	0.15 (0.61)	0.77
MSFT	1.16 (0.28)	0.99 (0.30)	0.50
TSLA	0.92 (0.19)	0.62 (0.60)	0.87
Mean	1.10	0.64	1.03

TABLE X
EXPERIMENT 2 RESULTS: MAXIMUM DRAWDOWN

Ticker	LLM+RL (σ)	RL-Only (σ)	LLM-Only
AAPL	0.29 (0.20)	0.45 (0.01)	0.28
AMZN	0.26 (0.12)	0.19 (0.14)	0.34
GOOGL	0.28 (0.06)	0.25 (0.18)	0.35
META	0.35 (0.11)	0.45 (0.27)	0.30
MSFT	0.19 (0.08)	0.17 (0.09)	0.21
TSLA	0.46 (0.05)	0.65 (0.13)	0.59
Mean	0.31	0.36	0.35

The hybrid agent did not consistently minimize MDD per stock but achieved values close to the best across agents, with the lowest overall mean (0.31). This suggests overall smoother drawdowns under uncertainty across the universe

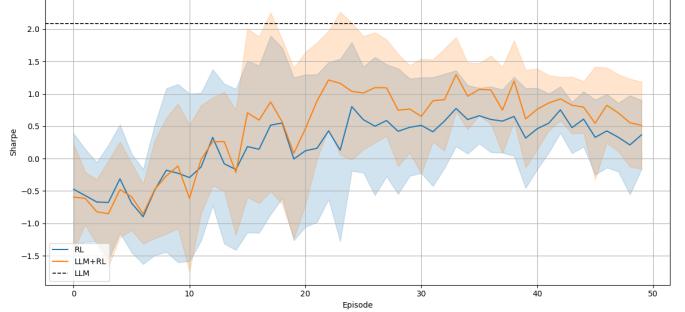


Fig. 3. Training Behavior for AAPL: Sharpe Ratio.

(see Tables IX and X).

IV. CONCLUSION AND FUTURE WORK

This study has explored an RL+LLM hybrid architecture for algorithmic trading, where LLMs generate guidance for RL agents to act as tactical executors.

Experiment 1 has shown that well engineered prompts improve the LLM's performance, with Prompt 4 achieving the highest SR and lowest uncertainty. Expert evaluations confirmed the rationale of generated strategies within the domain.

Experiment 2 has demonstrated that an RL agent guided by LLM signals outperforms the RL-only baseline in four out of

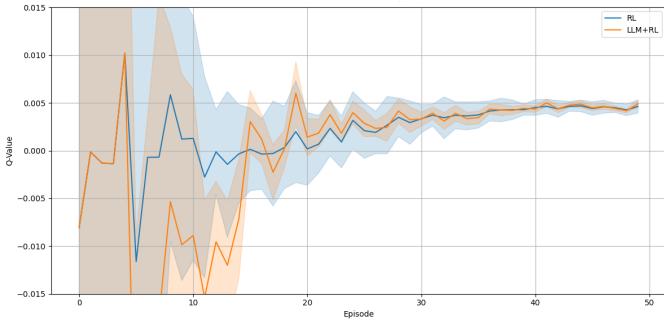


Fig. 4. Training Behavior for AAPL: Q-Values for LONG.

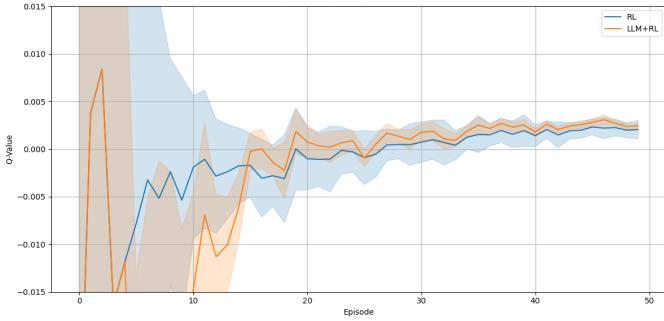


Fig. 5. Training Behavior for AAPL: Q-Values for SHORT.

six stocks when evaluated by their Sharpe Ratio. While MDD was not consistently reduced, the overall drawdowns remained low on average. Importantly, the underlying RL architecture was not modified; all observed improvements stemmed from LLM guidance.

Future research should address two main directions. First, while the LLM can guide the RL, reward shaping is necessary to attain optimal results. Second, modular specialization through multiple LLM agents prompted for specific domains may enable a mixture-of-experts architecture, and lessen the risk of confabulation.

Overall, this work presents a novel LLM+RL system that improves both return and risk outcomes. It supports modular, agentic setups where LLMs operate as trustworthy planners in financial decision making.

SUPPLEMENTARY MATERIAL

Full prompt templates (strategy and analyst), labeling heuristic pseudocode, extended dataset schema, and complete replication tables are available from the corresponding author upon request.

ACKNOWLEDGMENT

We thank the expert reviewers who contributed their time and expertise to this work.

REFERENCES

- [1] M. Chlistalla, “High-frequency trading: Better than its reputation?” Deutsche Bank Research, Frankfurt am Main, Germany, Tech. Rep. Research Briefing, Feb. 2011. [Online]. Available: <https://www.finextra.com/finextra-downloads/featuredocs/prod0000000000269468.pdf>
- [2] E. F. Fama and K. R. French, “A five-factor asset pricing model,” *Journal of Financial Economics*, vol. 116, no. 1, pp. 1–22, 2015.
- [3] T. Théate and D. Ernst, “An application of deep reinforcement learning to algorithmic trading,” *Expert Systems with Applications*, vol. 173, p. 114632, Jul. 2021.
- [4] S. M. Bartram, J. Branke, and M. Motahari, *Artificial Intelligence in Asset Management*. CFA Institute Research Foundation, 2020.
- [5] R. S. Sutton and A. G. Barto, *Reinforcement Learning: An Introduction*, 2nd ed., ser. Adaptive Computation and Machine Learning series. MIT Press, 2018. [Online]. Available: <http://incompleteideas.net/book/the-book-2nd.html>
- [6] X.-Y. Liu, H. Yang, J. Gao, and C. D. Wang, “Finrl: Deep reinforcement learning framework to automate trading in quantitative finance,” in *Proceedings of the Second ACM International Conference on AI in Finance*. ACM, Nov. 2021.
- [7] M. M. L. de Prado, “Beyond econometrics: A roadmap towards financial machine learning,” *Econometric Modeling: Theoretical Issues in Microeconomics eJournal*, 2019. [Online]. Available: <https://api.semanticscholar.org/CorpusID:199365784>
- [8] L. Onozo, F. Arthur, and B. Gyires-Tóth, “Leveraging LLMs for financial news analysis and macroeconomic indicator nowcasting,” *IEEE Access*, 2024, early Access. Online. Accessed: Feb. 10, 2025.
- [9] W. Zhang *et al.*, “A multimodal foundation agent for financial trading: Tool-augmented, diversified, and generalist,” New York, NY, USA, p. 4314–4325, 2024.
- [10] M. Wang, K. Izumi, and H. Sakaji, “LLMFactor: Extracting profitable factors through prompts for explainable stock movement prediction,” 2024. [Online]. Available: <https://arxiv.org/abs/2406.10811>
- [11] S. Schulhoff *et al.*, “The prompt report: A systematic survey of prompting techniques,” 2024. [Online]. Available: <https://arxiv.org/abs/2406.06608>
- [12] Y. Yu *et al.*, “Finmem: A performance-enhanced lilm trading agent with layered memory and character design,” in *Proceedings of the AAAI Spring Symposium Series*, R. P. A. Petrick and C. W. Geib, Eds. AAAI Press, Jan. 2024, pp. 595–597.
- [13] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, “Deep reinforcement learning: A brief survey,” *IEEE Signal Processing Magazine*, vol. 34, no. 6, pp. 26–38, 2017.
- [14] A. Lopez-Lira and Y. Tang, “Can chatgpt forecast stock price movements? return predictability and large language models,” 2023.
- [15] J. Huang and K. C.-C. Chang, “Towards reasoning in large language models: A survey,” 2022. [Online]. Available: <https://arxiv.org/abs/2212.10403>
- [16] M. Lopez de Prado, “The 10 reasons most machine learning funds fail,” *The Journal of Portfolio Management*, Tech. Rep., 06 2018.
- [17] L. Takara, A. Santos, V. Mariani, and L. Coelho, “Deep reinforcement learning applied to a sparse-reward trading environment with intraday data,” *Expert Systems with Applications*, vol. 238, p. 121897, 2024.
- [18] A. Chaddha and S. Yadav, “Examining the predictive power of moving averages in the stock market,” *Journal of Student Research*, vol. 11, no. 3, 2022.
- [19] S. Wang, H. Yuan, L. M. Ni, and J. Guo, “Quantagent: Seeking holy grail in trading by self-improving large language model,” 2024. [Online]. Available: <https://arxiv.org/abs/2402.03755>
- [20] A. Lopez-Lira, Y. Tang, and M. Zhu, “The memorization problem: Can we trust LLMs’ economic forecasts?” 2025. [Online]. Available: <https://arxiv.org/abs/2504.14765>
- [21] H. Gonen, S. Iyer, T. Blevis, N. Smith, and L. Zettlemoyer, “Demystifying prompts in language models via perplexity estimation,” in *Findings of the Association for Computational Linguistics: EMNLP 2023*, H. Bouamor, J. Pino, and K. Bali, Eds. Singapore: Association for Computational Linguistics, Dec. 2023, pp. 10136–10148.
- [22] A. Kaltchenko, “Entropy heat-mapping: Localizing GPT-based OCR errors with sliding-window shannon analysis,” 2025. [Online]. Available: <https://arxiv.org/abs/2505.00746>
- [23] L. M. Demajo, V. Vella, and A. Dingli, “Explainable AI for interpretable credit scoring,” 2020. [Online]. Available: <https://arxiv.org/abs/2012.03749>
- [24] G. Yona, R. Aharoni, and M. Geva, “Can large language models faithfully express their intrinsic uncertainty in words?” 2024. [Online]. Available: <https://arxiv.org/abs/2405.16908>

APPENDIX A

STRATEGY PROMPT

The final tuned prompt from Experiment 1 and the LLM strategy generator for Experiment 2, is available in 1.

Listing 1. Tuned Strategy Prompt

```

1 User_Context:
2   Last_Strategy_Used_Data:
3     last_returns: "{Last_LLM_Strat_Returns}"
4     last_action: "{Last_LLM_Strat_Action}"
5   Rationale: |
6     """{Last_LLM_Strat}"""
7
8 Stock_Data:
9   General:
10    Beta: {Market_Beta}
11    Classification: {classification}
12
13 Last_Weeks_Price:
14   Close: "{Close}"
15   Volume: "{Volume}"
16
17 Weekly_Past>Returns: "{Weekly_Past>Returns}"
18
19 Historical_Volatility:
20   HV_Close: "{HV_Close}"
21
22 Implied_Volatility:
23   IV_Close: "{IV_Close}"
24
25 Fundamental_Data:
26   Ratios:
27     Current_Ratio: "{Current_Ratio}"
28     Quick_Ratio: "{Quick_Ratio}"
29     Debt_to_Equity_Ratio: "{Debt_to_Equity_Ratio}"
30     PE_Ratio: "{PE_Ratio}"
31
32 Margins:
33   Gross_Margin: "{Gross_Margin}"
34   Operating_Margin: "{Operating_Margin}"
35   Net_Profit_Margin: "{Net_Profit_Margin}"
36
37 Growth_Metrics:
38   EPS_YoY: "{EPS_YoY_Growth}"
39   Net_Income_YoY: "{Net_Income_YoY_Growth}"
40   Free_Cash_Flow_YoY: "{Free_Cash_Flow_Per_Share_YoY_Growth}"
41
42 Technical_Analysis:
43   Moving_Averages:
44     20MA: "{20MA}"
45     50MA: "{50MA}"
46     200MA: "{200MA}"
47   MA_Slopes:
48     20MA_Slope: "{20MA_Slope}"
49     50MA_Slope: "{50MA_Slope}"
50     100MA_Slope: "{100MA_Slope}"
51     200MA_Slope: "{200MA_Slope}"
52   MACD:
53     Value: "{MACD}"
54     Signal_Line: "{Signal_Line}"
55     MACD_Strength: {MACD_Strength}
56   RSI:
57     Value: "{RSI}"
58   ATR: "{ATR}"
59
60 Macro_Data:
61   Macro_Indices:
62     SPX:
63       Close: "{SPX_Close}"
64       Close_20MA: "{SPX_Close_MA}"
65       Close_Slope: "{SPX_Close_Slope}"
66     VIX:
67       Close: "{VIX_Close}"
68       Close_20MA: "{VIX_Close_MA}"
69       Close_Slope: "{VIX_Close_Slope}"
70
71 Economic_Data:
72   GDP_QoQ: "{GDP_QoQ}"
73   PMI: "{PMI}"
74   Consumer_Confidence_QoQ: "{Consumer_Confidence_QoQ}"
75   M2_Money_Supply_QoQ: "{M2_Money_Supply_QoQ}"
76
77 PPI_YoY: "{PPI_YoY}"
78 Treasury_Yields_YoY: "{Treasury_Yields_YoY}"
79
80 Options_Data:
81   Put_IV_Skews:
82     OTM_Skew: "{OTM_Skew}"
83     ATM_Skew: "{ATM_Skew}"
84     ITM_Skew: "{ITM_Skew}"
85
86 20Day_Moving_Averages:
87   OTM_Skew_MA: "{MA_OTM_Skew}"
88   ATM_Skew_MA: "{MA_ATM_Skew}"
89   ITM_Skew_MA: "{MA_ITM_Skew}"
90
91 News_Sentiment: {news_sentiment}
92 News_Impact_Score: {news_impact_score}
93
94 System_Context(System):
95   Persona: {persona}
96   Portfolio_Objectives: {portfolio_objectives}
97   Instructions: |
98     Develop a LONG or SHORT trading strategy for a
99       single stock only for the next Month that
100      aligns with the 'portfolio_objectives'.
101      Follow these guidelines:
102
103 1. Stock Analysis:
104   - Evaluate price trends: Compare the Close
105     price against 20MA, 50MA, and 200MA to
106       assess momentum or reversals.
107   - Analyze returns: Use Weekly Past Returns to
108     validate trend sustainability.
109   - Contextualize volatility: Align 'HV_Close'
110     and 'HV_High' with recent price action
111       for trend validation.
112   - Incorporate beta: Use 'beta' to gauge
113     sensitivity to market movements.
114   - ICL Example: "Close_price_above_20MA_and_50
115     MA_with_stEEP_20MA_slope_signals_bullish_
116     momentum._Weekly_returns_confirm_a_
117     sustainable_uptrend."
118
119 2. Technical Analysis:
120   - Use RSI: Identify momentum signals (>70
121     overbought; <30 oversold) and divergences
122       for reversals.
123   - Validate with 'MACD': Use crossovers of 'MACD.Value' and 'Signal_Line', and 'MACD_Strength' for directional confidence
124
125   - Leverage 'RSI.value' divergences, and steep
126     'Moving_Averages' slopes. Or focus on
127       stable 'Moving_Averages' patterns on
128       stable historical volatility 'HV_Close'.
129   - ICL Example: "RSI_at_65,_a_positive_MACD_
130     crossover_indicate_bullish_momentum."
131
132 3. Fundamental Analysis:
133   - Evaluate growth metrics: Use 'EPS_YoY', 'Net_Income_YoY', and 'Free_Cash_Flow_YoY' for profitability and sustainability.
134   - Prioritize ratios: Low 'Debt_to_Equity_Ratio' and 'Current_Ratio' reflect financial stability.
135   - Focus on aggressive 'Growth Metrics' and earnings news.
136   - ICL Example: "EPS_YoY_growth_of_25%_and_low_
137     _Debt-to-Equity_ratio_of_0.5_support_
138     strong_financial_health_aligning_with_a_
139     LONG_strategy."
140
141 4. Macro Analysis:
142   - Align with market sentiment across 'Macro_Data':
143     - "SPX_Close_Slope_>_0_&&_VIX_Close_Slope_<_
144       _0": Bullish (Risk-On)
145     - "SPX_Close_Slope_<_0_&&_VIX_Close_Slope_>_
146       _0": Bearish (Risk-Off)
147   - Validate with 'Economic_Data':
148     - "GDP_QoQ_>_0_&&_PMI_>_50" leads to
149       Economic Expansion
150     - "'Treasury_Yields_YoY'_<_0" Signals
151       Recession Risk, especially if already

```

mentioned in 'Rationale'.
 123
 124 - ICL Examples:
 125 - "'SPX_Close_Slope' > 0 && 'VIX_Close_Slope'
 126 ' < 0 We have Market Confidence"
 127 - "'GDP_QoQ' Falling & 'PMI' < 50 We have
 128 an Economic Slowdown."
 129
 130 5. Options Analysis:
 131 - Compare 'OTM_Skew', 'ATM_Skew', and 'ITM_Skew' IV Skews: Assess differences to gauge market sentiment and directional bias using their '20Day_Moving_Averages'.
 132 - Leverage IV spikes to capitalize on speculative directional trades.
 133 - Example: "Rising ATM_Skew_MA > 0, market pricing_up move, with_stable_HV supports a LONG position, as_it indicates growing upside_expectations without_excessive_fear ."
 134
 135 6. News Analysis:
 136 - Use 'News_Sentiment' and 'News_Impact_Score' (1-3).
 137 - Only strong directional news (score = 3) should override other signals.
 138 - Medium news (score = 2) supports but does not lead.
 139 - Always check if news contradicts macro or technical trend.
 140
 141 7. Performance Reflection and Strategic Adaptation:
 142 - If 'Last_Strategy_Used_Data' is available:
 143 - Assess the outcome of the previous strategy by examining 'last_returns' and the chosen 'last_action'.
 144 - Determine if the result aligns with the expectations outlined in the previous 'Rationale'.
 145 - Identify if the direction (LONG or SHORT) led to desirable or undesirable outcomes.
 146 - You must NOT reuse or copy the previous 'Rationale'. It is only context for reflection.
 147 - Summarize in 1-2 sentences whether the previous strategy performed as expected.
 148 - Example: "The previous LONG strategy yielded positive returns, confirming the bullish setup based on RSI and moving averages."
 149 - Do NOT include language or phrasing from the previous rationale.
 150 - Confidence assignment:
 151 - Assign a Likert score (1 to 3) to your 'action_confidence':
 152 - 1: Low confidence; contradictory or weak alignment across features.
 153 - 2: Moderate confidence; partial alignment with moderate evidence.
 154 - 3: High confidence; strong convergence across key features.
 155 - Feature Attribution:
 156 - Rank the importance of each major feature used in your current rationale using a Likert scale (1 to 3):
 157 - 1: Minimal contribution; not required for the decision.
 158 - 2: Moderate contribution; relevant but not critical.
 159 - 3: High contribution; pivotal to the trading decision.
 160
 161 Output:
 162 action: Str. LONG or SHORT.
 163 action_confidence: int. Likert scale (1-3)
 164 confidence in the proposed 'action', adjusted
 165 based on prior strategy outcome if 'Last_Strategy_Used_Data' is available.
 166
 167 explanation: >

A concise rationale (max 350 words) justifying the proposed 'action'.
 168
 169 Include:
 170 - The top 5 weighted features used in the decision, each labeled with its Likert importance (1-3).
 171 (e.g., "Stock_Data.Price.Close,_Weight_3,_Technical_Analysis.RSI.Value,_Weight_1,_Options_Data.ATM_Skew,_Weight_2")
 172 - A reflective assessment of 'Last_Strategy_Used_Data', including whether the past 'action' was successful and was it maintained given prior 'Rationale'.
 173
 174 features_used:
 175 - feature: the features used from the prompt's context.
 176 direction: LONG, SHORT, or NEUTRAL
 177 weight: A Likert score (1 to 3) described in Feature Attribution.

APPENDIX B ANALYST PROMPT

The Analyst prompt used in Experiment 1 is presented in Listing 2, adapted from [10]. News corpora were anonymized prior to prompting.

Listing 2. Analyst Prompt

```

1 User_Context:  

2   Monthly_News_Articles_List: |  

3     "(articles_list)"|  

4  

5 System_Context:  

6   Persona: Financial Market Analyst  

7   Instructions: |  

8     Extract the 'Top 3' news factors influencing  

      stock price movements from the '  

      Monthly_News_Articles_List'. Follow these  

      steps:  

9  

10    1. Rank the news by relevance to stock price  

      movements:  

11      - Prioritize news related to significant  

        financial or market impacts (e.g.,  

        acquisitions, partnerships, guidance  

        revisions).  

12      - Weigh industry trends, macroeconomic  

        influences, and analyst ratings based on  

        their expected effect on the company  

        valuation.  

13      - News with broad or long-term implications  

        ranks higher.  

14  

15    2. Summarize content into key factors and  

      corporate events affecting stock prices,  

      using concise language and causal  

      relationships.  

16  

17    3. For each factor, assign:  

18      - 'Sentiment': +1 for positive, -1 for  

        negative, 0 for neutral or mixed  

19      - 'Market_Impact_Score': Likert scale from 1  

        to 3, where:  

20        - 1 = minimal relevance  

21        - 2 = moderate influence  

22        - 3 = high impact driver  

23  

24 Examples of factors influencing stock prices  

25   include:  

26     - Strategic partnerships or competitor  

        activity.  

27     - Industry trends or macroeconomic influences.  

28     - Product launches or market expansions.  

29     - Analyst ratings, significant stock price  

        moves, or expectations.  

30     - Corporate events: guidance revisions,  

        acquisitions, contracts, splits,  

        repurchases, dividends.  

31  

32 Example:
  
```

Algorithm 1: Expert Trade Heuristic

Data: Time-indexed price series
Result: Trade action: LONG (1) or SHORT (0)

```

1 foreach date t in dataset do
2   Pt ← Close(t);
3   r(10) ←  $\frac{P_{t+10}}{P_t} - 1$ , r(20) ←  $\frac{P_{t+20}}{P_t} - 1$ ;
4   rweighted ← 0.4 · r(10) + 0.6 · r(20);
5   if rweighted >= 0 then
6     Action ← LONG (Trade_Action = 1);
7   else
8     Action ← SHORT (Trade_Action = 0);

```

```

32   'A major tech company partners with a leading
33   automotive firm for EV battery innovation.
34   Analysts predict this could boost
35   revenues significantly.'
36
37   Ranked Factors:
38   1. factor: Strategic partnership in EV
      battery technology expected to increase
      revenue.
39     sentiment: +1
40     market_impact: 3
41   2. factor: Positive sentiment driven by
      projected long-term gains.
42     sentiment: +1
43     market_impact: 2
44   3. factor: Growing demand for EV technology
      anticipated to support future earnings.
45     sentiment: +1
46     market_impact: 2
47
48   Output:
49   factors:
      - factor: str. Summary of the news item. Max 70
        words.
      - sentiment: int. One of Positive +1, Negative
        -1, or Neutral 0
      - market_impact: int. Likert scale 1 to 3

```

APPENDIX C ALGORITHMS

The labeling algorithm emulates expert trading behavior by deliberately leveraging future return information to assign proxy trade actions in hindsight. This approach offers a cost-effective and scalable addition to manual annotation, capturing the general direction an informed trader might take. These synthetic labels are then provided to the LLM, along with a smaller set of HITL annotated examples.

APPENDIX D DATASET

Market Data

This market data (S_{mk}) included OHLCV price series as well as macro-level indicators and forward-looking sentiment signals. Specifically, it comprised:

- Daily returns of the S&P 500 Index (SPX) and NASDAQ-100 Index (NDX). These are market and sector indices,
- Implied Volatility (IV) and Historical Volatility (HV) metrics, derived from the stock's derivatives,
- The CBOE Volatility Index (VIX) as a proxy for market fear and option market expectations,
- *Weekly Past Returns*, which record the percentage change over the past four weekly intervals. The four-week

span was selected empirically to align with the model's monthly strategy generation frequency.

These features help in modeling short-term market dynamics.

Fundamental Data

Fundamental data (S_{fund}) has firm-level fundamentals and macroeconomic indicators. Macroeconomic variables provided contextual narrative for interpreting observed signals, and supporting regime identification [8], [9]. This set covered:

- **Liquidity ratios:** Current Ratio, Quick Ratio;
- **Leverage and coverage:** Debt-to-Equity, Interest Coverage;
- **Profitability metrics:** Gross Margin, Operating Margin, Return on Equity (ROE), Return on Assets (ROA);
- **Valuation:** Price-to-Earnings (P/E), Price-to-Book (P/B), Enterprise Value (EV), and Earnings Before Interest, Taxes, Depreciation, and Amortization (EBITDA).
- **Growth:** Revenue and Earnings Growth;
- **Macroeconomic indicators:** Gross Domestic Product (GDP), Purchasing Managers' Index (PMI), Producer Price Index (PPI), Consumer Confidence Index (CCI), U.S. 10-Year Treasury Yield, and the 10Y–2Y yield curve slope.

To enhance temporal abstraction, all variables were computed as quarter-over-quarter (QoQ) or year-over-year (YoY) percentage changes. It is critical to take first-order dynamics as LLMs can recall absolute numbers for economic details, allowing look-ahead bias in the backtests [20].

Analytics

Technical indicators (S_{an}) were computed over rolling 20-day windows using the open-source TA-Lib⁷ library. These features include:

- Simple Moving Averages (SMA) over 20, 50, 100, 200 trading-day horizons,
- Relative Strength Index (RSI),
- Average True Range (ATR) for volatility,
- Moving Average Convergence Divergence (MACD) with its signal line and derived strength,
- Volume-Weighted Average Price (VWAP) as a reference anchor for intraday valuations.

Each indicator was extended with slope and z-score to assist the LLM in capturing directional shifts and the statistical significance of deviations. These technical indicators are widely used in trading practice and academic research [18].

Alternative Data

Structured representations of financial news headlines (S_{alt}) were extracted using a large language model (LLM), which anonymized and synthesized the content into latent factors. Following the LLMFactor methodology [10], each news item was distilled into 2–5 interpretable factors, capturing macroeconomic and firm-specific signals.

⁷<https://ta-lib.org/>

Instrument	Paper SR	SR ($\pm\sigma$) [p -value]	MDD ($\pm\sigma$)
AB InBev	0.187	1.21 (0.30) [0.00]	0.18 (0.08)
Alibaba	0.021	0.06 (0.02) [0.00]	0.09 (0.01)
Amazon	0.419	0.39 (0.45) [0.85]	0.30 (0.09)
Apple	1.424	1.19 (0.55) [0.22]	0.29 (0.09)
Baidu	0.080	0.20 (0.17) [0.00]	0.36 (0.09)
CCB	0.202	0.33 (0.25) [0.04]	0.24 (0.14)
Coca Cola	1.068	1.07 (0.53) [0.50]	0.25 (0.04)
Dow Jones	0.684	0.70 (0.30) [0.91]	0.25 (0.05)
ExxonMobil	0.098	0.10 (0.35) [0.91]	0.34 (0.08)
FTSE 100	0.103	0.50 (0.23) [0.00]	0.31 (0.08)
Google	0.227	-0.54 (0.59) [0.00]	0.43 (0.13)
HSBC	0.011	0.38 (0.17) [0.00]	0.29 (0.05)
JPMorgan Chase	0.722	0.72 (0.31) [0.98]	0.26 (0.06)
Kirin	0.852	0.85 (0.42) [0.99]	0.39 (0.07)
Meta	0.151	0.63 (0.61) [0.01]	0.45 (0.27)
Microsoft	0.987	0.70 (1.00) [0.38]	0.28 (0.16)
NASDAQ 100	0.845	0.85 (0.35) [1.00]	0.16 (0.05)
Nikkei 225	0.019	0.26 (0.29) [0.02]	0.29 (0.07)
Nokia	-0.094	0.07 (0.24) [0.00]	0.57 (0.15)
PetroChina	0.156	0.22 (0.29) [0.29]	0.67 (0.00)
Philips	0.675	1.40 (0.50) [0.00]	0.25 (0.03)
S&P 500	0.834	0.83 (0.25) [1.00]	0.14 (0.04)
Shell	0.425	0.42 (0.37) [0.95]	0.51 (0.05)
Siemens	0.426	0.39 (0.23) [0.43]	0.26 (0.12)
Sony	0.424	0.42 (0.36) [0.97]	0.16 (0.04)
Tesla	0.621	0.48 (0.41) [0.29]	0.52 (0.09)
Tencent	-0.198	-0.19 (0.33) [0.98]	0.10 (0.09)
Toyota	0.304	0.36 (0.27) [0.37]	0.45 (0.10)
Volkswagen	0.216	0.45 (0.18) [0.00]	0.48 (0.09)

TABLE XI
REPLICATION METRICS FOR [3]

To mitigate memorization and data leakage risks, named entities and dates were anonymized (e.g., “Tesla” becomes “the Company”).

APPENDIX E REPLICATED BENCHMARK METRICS

We report the replicated benchmark metrics in Appendix E for the assets used in [3]. We include the mean SR and MDD, each averaged across 25 runs with standard deviation σ .

For the SR, we conduct a two-sided one-sample t -test to assess whether the metric is significantly different from the published value. The null hypothesis H_0 assumes equivalence, i.e. $H_0 : \mu_{SR} = SR_{paper}$.

Since this is a replication test, failing to reject H_0 indicates successful replication. p -values are computed only for SR; other metrics are reported without significance testing.

All assets have been successfully replicated within acceptable bounds, with exceptions highlighted in bold. Notably, GOOGL, one of the stocks included in our test environment, exhibited a statistically significant deviation from the original benchmark, with a p -value below 0.05.