Calibrated Signals @ FinMMEval 2026 Task 3: Where Quant Meets Reasoning: A Multi-Agent Framework for Trading Intelligence*

Gaurav Jain[1,*,†], Hitesh Hitesh[1,†], Sachit Bhardwaj[1,†], Himanshu Behl[1,†], Sparsh Sundriyal[1,†], Prabal Gautam[1,†] and Steven Zimmerman[2,†]

[1]SBO Bangalore
[2]SHELL AUSTRIA GESELLSCHAFT M.B.H

Abstract

We present a multi-agent system for daily trading decision making submitted to Task 3 of CLEF FinMMEval Lab 2026. The system combines a deterministic rule-based quantitative engine computing 20+ technical indicators across six categories — trend, momentum, volatility, volume, risk, and pattern recognition — with three LLM-powered specialist agents for news sentiment analysis, SEC filings forensics, and final decision synthesis. A three-phase orchestration runs live market enrichment first (Phase 1), then three analysis agents in parallel (Phase 2), before a Howard Marks-inspired synthesiser produces a BUY / HOLD / SELL decision (Phase 3). The system is deployed on Azure Functions and comfortably fits within the 3-minute evaluation budget, with observed end-to-end latency of 35–110 seconds. We evaluated on TSLA equity and BTC cryptocurrency backtesting data using Cumulative Return as the primary metric, with Sharpe Ratio and Maximum Drawdown as secondary risk-adjusted measures. A signal agreement analysis quantifies when the rule-based and LLM agents concur or conflict, and an expected ablation study outlines the hypothesised contribution of each agent.

Keywords

multi-agent systems, hybrid quantitative-LLM system, financial decision making, technical analysis, signal fusion, trading intelligence, FinMMEval, CLEF 2026

1. Introduction

Financial markets generate heterogeneous signals continuously: numerical price histories, order-flow volume, real-time news, regulatory disclosures, analyst recommendations, and macro indicators. Human portfolio managers synthesise these signal modalities intuitively, weighting each according to the prevailing market regime, asset class, and time horizon. Replicating this synthesis computationally requires a system capable of reasoning simultaneously over structured quantitative data *and* unstructured natural language — a challenge at the intersection of financial signal processing and natural language understanding.

Task 3 of the FinMMEval Lab (CLEF 2026) [1, 2] operationalises this challenge as a daily trading game: each participating endpoint receives a JSON payload containing the current price, recent news articles, a momentum label, a 20-point synthetic price history, and optionally 10-K/10-Q filing excerpts. The endpoint must return a single decision — BUY, HOLD, or SELL — within three minutes. Positions are executed at daily close prices and evaluated primarily on Cumulative Return over the evaluation window. Two structurally distinct assets are included: TSLA (a high-beta US equity with full SEC filing coverage) and BTC (a 24/7 cryptocurrency with no regulatory filings and extreme volatility).

CLEF 2026: Conference and Labs of the Evaluation Forum, 2026
*System description paper submitted to the FinMMEval Lab Working Notes at CLEF 2026.
*Corresponding author.
†These authors contributed equally.
g.jain3@shell.com (G. Jain); hitesh.hitesh@shell.com (H. Hitesh); Sachit.Bhardwaj@shell.com (S. Bhardwaj); Himanshu.Behl@shell.com (H. Behl); Sparsh.Sundriyal@shell.com (S. Sundriyal); Steven.Zimmerman@shell.com (S. Zimmerman)
0009-0006-9451-0061 (G. Jain); 0000-0001-9061-4388 (H. Hitesh); 0000-0003-0644-7500 (S. Bhardwaj); 0009-0000-0727-8185 (H. Behl); 0009-0003-7983-5704 (S. Sundriyal); 0000-0002-4686-2066 (S. Zimmerman)

This dual-asset design tests both the breadth and the graceful-degradation properties of any submitted system.

A naïve approach — relying solely on the sparse 20-point synthetic history — discards rich data available from live market APIs. A purely LLM-based approach lacks the precision and reproducibility of rule-based technical analysis. Our hypothesis is that **a hybrid system anchoring LLM reasoning with deterministic quantitative signals outperforms either component in isolation**, because the quantitative layer provides an objective, statistically grounded baseline that the LLM can confirm, override, or contextualise with fundamental and sentiment information.

> *Core research question:* Can a hybrid system combining deterministic technical analysis with LLM-based qualitative reasoning outperform either approach in isolation for daily trading decisions across structurally different asset classes?

Contributions:

1. A phased multi-agent architecture with live market data enrichment in Phase 1, replacing the sparse synthetic payload history.
2. A rule-based quantitative engine computing 20+ technical indicators with mathematically principled composite scoring.
3. An LLM prompt engineering framework using persona grounding, annotated JSON schemas, chain-of-thought decomposition, and critical instruction repetition (Section 5).
4. Parallel agent execution via `ThreadPoolExecutor` with per-agent timeout caps and graceful degradation, fitting within the 3-minute budget.
5. A production deployment on Azure Functions with end-to-end latency profiling and analysis of the Finnhub headline-only trade-off.
6. An expected ablation analysis outlining the hypothesised marginal contribution of each agent to trading performance across TSLA and BTC.

This working notes paper prioritises system description, engineering decisions, and empirical observations from the live Azure Function deployment.

2. Related Work

2.1. Algorithmic Trading and Technical Analysis

Technical analysis uses historical price and volume data to forecast future price movements. The foundational reference [3] catalogues 50+ indicators across trend, momentum, and volume categories; subsequent work has focused on combining multiple indicators into composite scoring systems to reduce individual indicator false-positive rates. MACD, RSI, Bollinger Bands, and ADX are among the most widely tested indicators in algorithmic trading, with documented predictive value across multiple equity markets [3]. Volume-weighted indicators such as OBV and MFI capture order-flow information that pure price indicators miss. Our quantitative agent computes 20+ indicators across six categories and aggregates them via a weighted voting scheme.

2.2. LLM-Based Financial Agents

FinGPT [4] demonstrates that domain-specific fine-tuning on financial news yields strong sentiment classification. BloombergGPT [5] pre-trains from scratch on a 363B-token financial corpus. FinRobot [6] introduces a modular agent framework producing analyst-grade research reports. Despite these advances, existing LLM financial agents typically operate on a single modality. Our system fills this gap by fusing a deterministic quantitative engine (numerical OHLCV data) with three LLM agents (news text, SEC filings text, and structured JSON from all prior agents).

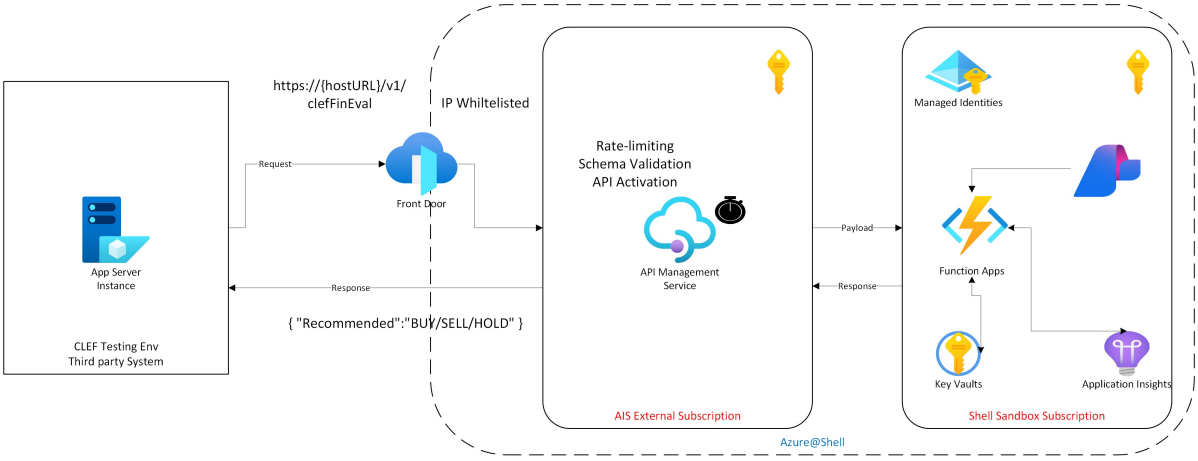

Figure 1: High Level Design

2.3. Multi-Agent Systems and Prompt Engineering

Multi-agent decomposition has been applied to portfolio management [7] and standardised agent benchmarking [8]. Our architecture differs by combining deterministic rule-based agents with generative LLM agents under a hard 3-minute latency constraint and evaluating across two structurally distinct asset classes simultaneously. Prompt engineering choices follow the empirical literature: chain-of-thought prompting [9] improves multi-step reasoning; annotated JSON schemas reduce format violations [10]; critical instruction repetition counteracts the lost-in-the-middle effect [11].

3. System Architecture

3.1. High Level Design

Figure 1 illustrates a secure and scalable cloud-based integration architecture implemented on Microsoft Azure to support our trading decision assistant workflow. The system is initiated by the CLEF testing environment, where an application server transmits requests to a publicly exposed API endpoint via Azure Front Door. This entry layer enforces network-level access control through IP whitelisting, thereby restricting interactions to authorised sources only.

Incoming requests are subsequently routed to an Azure API Management (APIM) layer hosted in a dedicated external subscription. Within this layer, governance and validation mechanisms—including rate limiting, schema validation, and controlled API exposure—are systematically applied to ensure compliance, reliability, and protection against malformed or excessive requests.

Validated requests are forwarded to backend Azure Function Apps deployed in a sandbox subscription. These serverless components encapsulate the core processing logic and interact with supporting services such as Azure Key Vault for secure secret management and Application Insights for monitoring and telemetry collection. Authentication between services is implemented using Managed Identities, thereby eliminating the need for explicit credential handling.

Upon completion of processing, responses are propagated back through the same communication pathway to the originating system. The output is delivered in a structured format, typically representing decision outcomes such as BUY, SELL, or HOLD recommendations. Overall, the architecture demonstrates a modular and resilient approach for integrating CLEF systems with enterprise cloud services, while emphasising security, scalability, and operational observability.

3.2. Multi Agent Orchestration

The system is a three-phase orchestrator deployed as a single Azure Function. All five agents execute as ordinary Python function calls inside a single process, eliminating inter-service HTTP overhead.

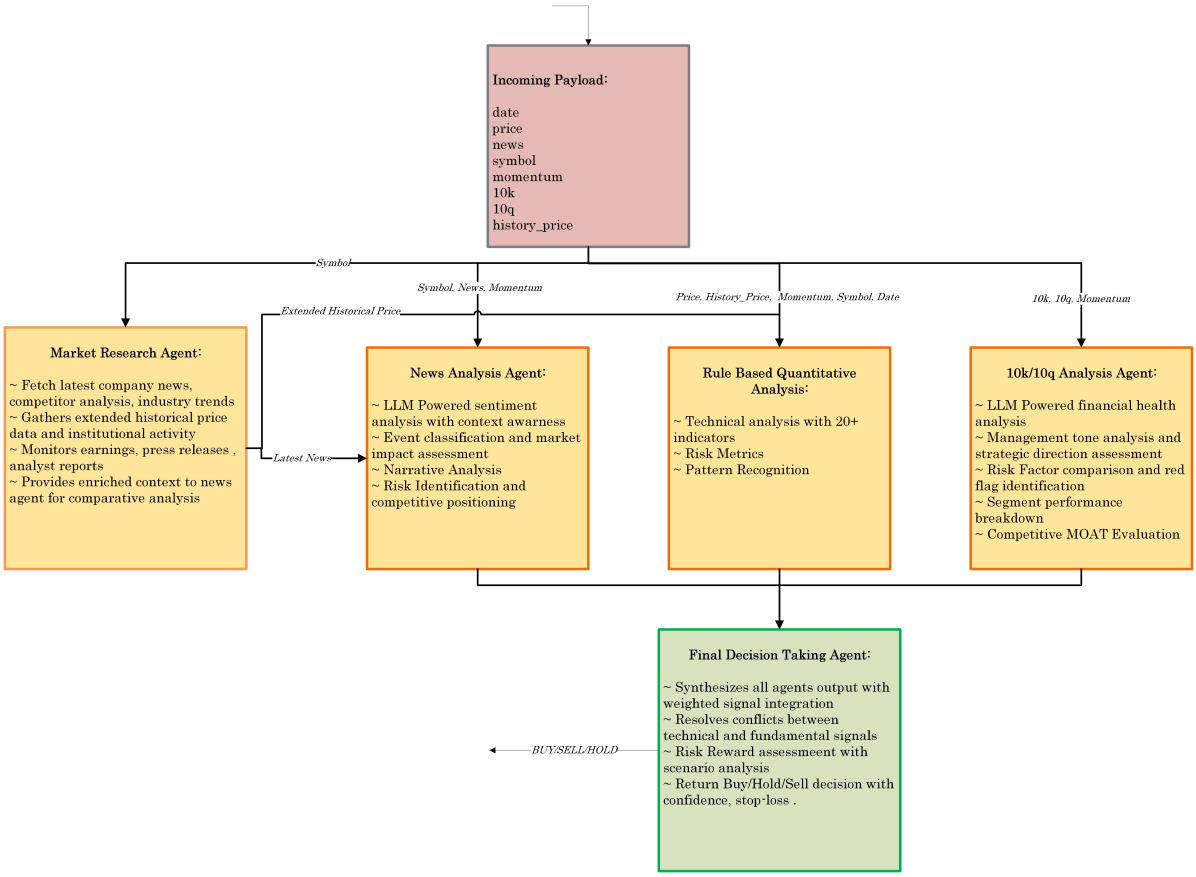

Figure 2: High-level Agent Orchestration. Phase 1 (Market Research Agent) enriches the payload with live 30-day OHLCV data before Phase 2 runs three analysis agents in parallel via `ThreadPoolExecutor`. Phase 3 (Final Answer Agent) synthesises all signals into a single BUY/HOLD/SELL decision.

3.3. Key Design Decisions

Phase 1: Market Research Agent fetches real 30-day OHLCV data from Finnhub *before* the quantitative agent runs, replacing the sparse 20-point synthetic payload history. Without this step, SMA-20, Bollinger Bands, and RSI are computed on fabricated data and their accuracy degrades substantially.

Phase 2: The three analysis agents share no data dependencies. Concurrent execution via `ThreadPoolExecutor(max_workers=3)` reduces Phase 2 wall-clock time to the *slowest single agent*, saving approximately 30–90 seconds over sequential execution.

Per-agent timeout and graceful degradation. Each Phase 2 future is capped at 150 seconds. A timed-out or empty-input agent returns `{"not_available": true}`. The Final Decision Taking Agent synthesises from whatever agents completed, redistributing signal weights proportionally.

BTC vs. TSLA routing. SEC Filings Agent is unconditionally skipped for BTC. Finnhub news uses `/news?category=crypto` for BTC and `/company-news` for TSLA.

Table 1
Production end-to-end latency breakdown (seconds). Phase 2 reports the *slowest* concurrent agent, not the sum.

Phase	Component	Min	Avg	Max
Phase 1	Finnhub API (6 parallel)	2	4	8
Phase 2	Quantitative (rule-based)	<1	<1	1
Phase 2	News Analysis LLM	10	28	55
Phase 2	SEC Filings LLM	10	30	60
Phase 2	**Wall-clock (max)**	**10**	**30**	**60**
Phase 3	Final Answer LLM	12	22	45
Total	**End-to-end**	**35**	**62**	**110**
Competition budget				*180 s*

Finnhub headlines-only trade-off. The Market Research Agent fetches Finnhub headlines and summaries, not full article bodies. Scraping full text would add 5–30 seconds of non-deterministic latency per request (rate limits, paywalls, JavaScript rendering). Headlines plus 1–3 sentence summaries are sufficient for LLM sentiment analysis: instruction-tuned models are trained on exactly this structure (headline + short abstract).

3.4. Concurrency Model

`ThreadPoolExecutor` is used in two places. In Phase 1, six independent Finnhub HTTP calls (company profile, real-time quote, 30-day OHLCV candles, financial metrics, analyst recommendations, news) run in parallel with `max_workers=3`. In Phase 2, the three analysis agents run concurrently. Python's GIL releases during I/O waits, so threads achieve true parallelism for network requests. Observed latency is summarised in Table 1.

4. Methodology

Our system is a **five-agent multi-signal reasoning pipeline** – two deterministic agents (market research and rule-based quantitative analysis) and three LLM agents (news sentiment, SEC filings, and final synthesis) — that processes each daily payload through specialised agents across three orchestration phases, then synthesises their outputs into a final Buy/Hold/Sell decision via an LLM-powered arbitration agent. Figure 2 illustrates the end-to-end architecture.

4.1. Task Definition

We address the *FinMMEval* shared task at CLEF 2026 [1, 2], which evaluates financial multimodal reasoning in a trading decision setting. Concretely, given a daily market snapshot $D_t = \{\text{price, news, momentum, history\_price, 10k, 10q}\}$ for asset s on trading day t, a system must produce a ternary action $a_t \in \{\text{Buy, Hold, Sell}\}$.

Each action implies a *full position change*: Buy opens a long, Hold retains a flat position, and Sell opens a short. A new action at time t fully replaces the position carried from $t-1$; trades are executed at the daily closing price with no explicit transaction costs. The full dataset comprises 273 daily observations per asset; unless stated otherwise, all results reported in this paper use a fixed 181-day evaluation subset spanning 2025-11-01 to 2026-04-30 (Section 4.3).

4.2. Input Payload

Each daily observation is delivered as a structured payload whose fields are summarised in Table 2. The payload combines *price signals* (spot price and a 20-point synthetic history), a *textual momentum label*

Table 2
Input payload fields for the FinMMEval trading task. Fields `10k`/`10q` are absent for BTC.

Field	Type	Description
`date`	string	Observation date (YYYY-MM-DD)
`symbol`	list	Asset ticker(s)
`price`	dict	Symbol → closing price
`news`	dict	Symbol → news text
`momentum`	dict	Symbol → {bullish / bearish / neutral}
`history_price`	dict	Symbol → list of 20 daily price records
`10k`	dict/null	Symbol → annual report excerpts
`10q`	dict/null	Symbol → quarterly report excerpts

Table 3
Sample rows from the official backtesting datasets. ΔP = `future_price_diff`. *Correct* is the theoretically optimal action given next-day price movement. **Bold** rows highlight cases where the momentum label disagrees with the optimal action.

Asset	Date	Price	Momentum	ΔP	Correct
TSLA	2025-08-01	\$302.63	bearish	+0.00	HOLD
TSLA	2025-08-02	\$302.63	bearish	+0.00	HOLD
TSLA	**2025-08-03**	**\$302.63**	**bearish**	**+6.63**	**BUY**
TSLA	2025-08-04	\$309.26	bullish	−0.54	SELL
BTC	2025-08-01	\$113,343	bearish	−558	SELL
BTC	**2025-08-02**	**\$112,785**	**bearish**	**+1,342**	**BUY**
BTC	**2025-08-03**	**\$114,127**	**bearish**	**+1,050**	**BUY**
BTC	2025-08-04	\$115,176	bullish	−985	SELL

derived from news sentiment, raw *news text*, and—for equity assets—*regulatory filings* (`10k` and `10q` excerpts).

4.3. Backtesting Datasets

Both datasets are drawn from the official `TheFinAI/CLEF_Task3_Trading` benchmark [1, 2, 8] and each provide 273 daily observations per asset beginning 1 August 2025. All results reported in Section 7 use a common 181-day evaluation subset spanning 2025-11-01 to 2026-04-30; earlier observations serve only for indicator warm-up and illustration (e.g. the sample rows in Table 3).

TSLA (equity). Tesla, Inc. closing prices open at \$302.63 on the first evaluation day. The full SEC filing suite (`10k`/`10q`) is available for every row, providing rich fundamental context alongside daily news and price history.

BTC (cryptocurrency). Bitcoin spot prices open above \$113,000. No regulatory filings exist for BTC; the system must rely exclusively on price, news, and momentum signals—a strictly harder sub-task.

Table 3 shows representative rows from both assets. A key observation is that the `momentum` label frequently *disagrees* with the theoretically optimal action: for BTC rows 2 and 3, the label is *bearish* yet the ground-truth optimal action is BUY. This motivates multi-source signal fusion rather than simple momentum following.

4.4. Evaluation Metrics

The primary metric is **Cumulative Return (CR)**:

$$\mathrm{CR} = \prod_{t=1}^{T}\big(1 + r_t\big) - 1, \tag{1}$$

where r_t is the daily return for the held position. Secondary metrics include: **Sharpe Ratio** ($\bar{r}/\sigma_r \times \sqrt{252}$), **Maximum Drawdown (MDD)**, **Daily Volatility**, and **Annualised Volatility**. A supplementary **Win Rate** is computed over non-HOLD rows as the fraction of directional calls where the predicted action agrees with the sign of `future_price_diff`.

4.5. Baselines

Buy-and-Hold. Long from day 1 throughout the evaluation window.

Hold-only. Always returns HOLD; zero risk and zero return.

Momentum-only. Mirrors the `momentum` label directly (*bullish*→BUY; *bearish*→SELL; *neutral*→HOLD).

4.6. Pipeline Overview

For each daily row the pipeline executes in two stages. Listing 1 shows the top-level orchestration.

Listing 1: Two-stage pipeline orchestration

```
def run_task3_pipeline(row: RequestModel) -> Dict[str, Any]:
    symbol  = row.symbol[0]
    price   = row.price[symbol]
    payload = _build_agent_payload(row)

    # Stage 1: market research (Finnhub API)
    market_research = research_company(symbol, row.date, price)
    # Extend history_price with real Finnhub daily prices
    daily_prices = market_research.get("daily_prices_1month", [])
    if daily_prices:
        payload["history_price"][symbol].extend(daily_prices)

    # Stage 2: quant, news, SEC filings (150 s timeout each)
    _AGENT_TIMEOUT = 150
    with ThreadPoolExecutor(max_workers=3) as executor:
        quant_f    = executor.submit(decide_action, payload)
        news_f     = executor.submit(analyze_news, ...)
        filings_f  = executor.submit(analyze_sec_filings, ...)
        quant_result    = quant_f.result(timeout=_AGENT_TIMEOUT)
        news_result     = news_f.result(timeout=_AGENT_TIMEOUT)
        filings_result  = filings_f.result(timeout=_AGENT_TIMEOUT)

    # Stage 3: final synthesis
    return synthesize_final_decision(
        symbol, row.date, price,
        quant_result, market_research,
        news_result, filings_result,
        momentum=row.momentum[symbol])
```

Stage 1 (Market Research) is serial because its Finnhub price data extends the history vector consumed by the Stage 2 quantitative agent. If any Stage 2 agent exceeds the 150-second timeout, it is replaced with a HOLD fallback so the pipeline always produces a response.

4.7. Agent Design

The orchestration comprises five agents. Agents 1 and 2 are fully deterministic (no LLM); Agents 3, 4, and 5 call Azure OpenAI `o4-mini` (`temperature=0`), deployed on Azure AI Foundry and accessed via the `AIProjectClient` SDK.

4.7.1. Agent 1: Market Research (Deterministic)

Issues six parallel Finnhub REST API calls per request: (i) company profile, (ii) real-time quote, (iii) 30-day OHLCV candles (resolution=D), (iv) financial metrics (P/E, EPS, beta, ROE, ROA, D/E ratio), (v) analyst consensus (buy/hold/sell/strongBuy/strongSell counts), and (vi) company news. Competitor news is fetched in a second pass once the Finnhub profile returns the industry string (e.g. `"Automobiles"`), improving relevance. Each call carries a 4-second individual timeout with silent fallback to empty data. The 30-day OHLCV array replaces `history_price` in the payload before Agent 2 runs.

The market research agent queries the Finnhub REST API to retrieve per-symbol fundamentals, analyst consensus, and up to 30 days of real closing prices (Listing 2).

Listing 2: Market research output structure (`market_research_agent.py`).

```
return {
    "company_summary": f"{company_name} ({symbol}) ...",
    "historical_context": {
        "52_week_high": week_52_high,
        "52_week_low":  week_52_low,
        "beta":         beta,
        "pe_ratio":     pe_ratio,
    },
    "analyst_sentiment": {
        "strong_buy": ..., "buy": ...,
        "hold": ..., "sell": ..., "strong_sell": ...,
    },
    "financial_metrics": {
        "roe": ..., "roa": ...,
        "profit_margin": ..., "debt_to_equity": ...,
    },
    "daily_prices_1month": historical_prices,  # appended to payload
    "company_news": finnhub_news_items,        # forwarded to news agent
}
```

For BTC, which is not covered as an equity by Finnhub, only publicly available price quotes and news headlines are returned; all equity-specific fields are marked as unavailable rather than used as substantive signals.

4.7.2. Agent 2: Rule-Based Quantitative Analysis (Deterministic)

Computes 20+ technical indicators from the enriched OHLCV history. All computations are fully deterministic; the same input always produces the same output.

Trend indicators. SMA-20/50 golden/death cross (score ± 0.5); EMA-12/26 position; MACD histogram sign (± 0.4); ADX trend strength (confidence factor 0.8 if ADX> 25, else 0.4). The SMA golden cross ($\text{SMA}_5 > \text{SMA}_{20}$) is the strongest single bullish signal in the composite; the MACD histogram captures momentum velocity simultaneously.

Momentum oscillators. RSI-14 overbought/oversold (∓ 0.3); Stochastic %K/%D-14 (∓ 0.2); Williams %R-14 (∓ 0.2); CCI-20 (∓ 0.2 for $|\text{CCI}| > 100$); ROC-12 (directional momentum).

Volatility indicators. Bollinger Bands-20 price proximity (∓ 0.3); ATR-14 for volatility regime context (high ATR reduces overall confidence scaling).

Volume indicators. OBV trend confirmation (± 0.3); MFI-14 volume-weighted RSI (∓ 0.3 for overbought/oversold); VWAP price-vs-average position.

Support/Resistance and Pattern. Classic pivot levels; Fibonacci retracement levels (23.6%, 38.2%, 50.0%, 61.8%) — proximity within 1% flagged as reversal zone (± 0.3).

Risk metrics. Rolling 30-day Sharpe ratio; rolling 30-day Maximum Drawdown. High drawdown on recent history is a risk flag forwarded to the Final Decision Taking Agent.

Composite score. $s_{\text{norm}} = s / s_{\max}$ maps to signal labels:

s_{norm} range	Label
> 0.30	Strong Bullish
$(0.10,\ 0.30]$	Weak Bullish
$[-0.10,\ 0.10]$	Neutral
$[-0.30,\ -0.10)$	Weak Bearish
≤ -0.30	Strong Bearish

The full JSON — every sub-signal name, computed value, and score contribution — is forwarded verbatim to Agent 5, providing complete transparency into the quantitative signal chain.

The quantitative agent is a pure rule-based engine with no LLM call. It consumes the extended price series (dataset history + Finnhub daily prices) and computes 20+ technical indicators across six categories, accumulating a signed scalar score s that drives a normalised signal $\hat{s} = \text{clip}(s/3.7,\ -1, +1)$. Listing 3 shows the scoring logic for two representative indicator groups.

Listing 3: Indicator scoring in the quantitative agent (`rule_based_quantitative_analysis.py`).

```
# --- Trend: SMA(5) vs SMA(20) crossover ---
if sma_5 > sma_20:
    trend_signals["sma_crossover"] = "bullish"
    score += 0.5          # bullish contribution
elif sma_5 < sma_20:
    trend_signals["sma_crossover"] = "bearish"
    score -= 0.5          # bearish contribution

# --- Momentum: RSI(14) ---
rsi = compute_rsi(series, window=14)
if rsi < 30:
    momentum_signals["rsi_signal"] = "oversold"
    score += 0.4
elif rsi > 70:
    momentum_signals["rsi_signal"] = "overbought"
    score -= 0.4

# --- Normalise ---
MAX_RAW_SCORE = 3.7       # sum of all indicator weights
normalized_score = max(-1.0, min(1.0, score / MAX_RAW_SCORE))
```

The full indicator set comprises: SMA(5/20/50), EMA(12/26), MACD with histogram crossover, ADX(14), linear-regression trend; RSI(14), Stochastic %K/%D(14), Williams %R(14), CCI(20), ROC(12), MFI(14); Bollinger Bands (20, 2σ), ATR(14); OBV, VWAP; Fibonacci retracement levels; and support/resistance proximity detection. The structured JSON output feeds the Final Answer Agent.

4.7.3. Agent 3: News Sentiment Analysis (LLM)

Fuses the task-supplied `news` field with live Finnhub headlines fetched in Phase 1. Skipped (returns `{"not_available": true}`) when both streams are empty. Persona: *elite financial news analyst* with six enumerated capability domains (market-moving event detection, cross-asset correlation, multi-timeframe sentiment, quantitative language analysis, risk-opportunity asymmetry, regulatory/macro awareness).

The structured output schema has nine top-level sections: `executive_summary`, `sentiment_analysis`, `event_classification`, `financial_impact_analysis`, `key_themes_and_narratives`, `risk_and_opportunity_assessment`, `competitive_and_market_context`, `market_reaction_prediction`, and `synthesis_and_recommendation`. Every numeric field is annotated with its expected value range inline (e.g. `# -1 to +1`) to enforce structured, quantified reasoning rather than narrative hedging [10].

```
{
  "sentiment_analysis": {
    "primary_score": 0.0,        # -1 (bearish) to +1 (bullish)
    "confidence": 0.0,           # 0 to 1
    "time_horizon": ""           # immediate|short|medium
  },
  "financial_impact": "",        # minor|moderate|major|critical
  "synthesis_and_recommendation": {
    "final_direction": "",       # bullish|bearish|neutral
    "confidence": 0.0
  }
}
```

The news agent receives three text inputs: (i) the dataset's `news` field, (ii) live Finnhub headlines from the market research stage, and (iii) the market-research context summary. An instruction-tuned LLM hosted on **Azure AI Foundry** is called via the `AIProjectClient` SDK and constrained to return a JSON object (Listing 4).

Listing 4: Required JSON output schema enforced in the news analysis prompt (`prompts.py`).

```
{
  "verdict":            "3-4_sentence_summary_...",
  "overall_sentiment":  "bullish|bearish|neutral|mixed",
  "expected_price_direction": "up|down|sideways",
  "confidence_level":   "very_high|high|medium|low|very_low",
  "sentiment_strength": "strongly_bullish|...|strongly_bearish",
  "event_type":         "earnings|regulatory|product_launch|...",
  "event_materiality":  "high|medium|low",
  "top_risks":          ["..."],
  "top_catalysts":      ["..."],
  "trading_bias": {
      "short_term":  "bullish|bearish|neutral",
      "medium_term": "bullish|bearish|neutral"
  }
}
```

4.7.4. Agent 4: SEC Filings Analysis (LLM)

Performs forensic analysis of 10-K/10-Q excerpts. Unconditionally skipped for BTC. Skipped for TSLA when both fields are null. Persona: *forensic financial analyst* with explicit expertise in accounting quality

Table 4
Hard constraints in the Final Decision Taking Agent system prompt.

Constraint	Description
Howard Marks persona	Second-level thinking, risk-before-return, cycle awareness. Prioritises downside before upside; demands calibrated confidence.
Reward/risk gate	reward_risk_ratio < 1.5 mandates HOLD regardless of signal direction.
Signal weights	Quantitative 30%, News 25%, Filings 25%, Market Research 20%. Missing agents redistribute weight proportionally.
Confidence tiers	All four aligned: 80–95%; three of four: 65–80%; two of four: 40–60% (likely HOLD); major conflicts: HOLD.
Five-step CoT [9]	`reasoning` field: data quality → per-agent verdicts → conflict resolution → risk-reward calc → final synthesis.

assessment. The word "forensic" primes the model to look for inconsistencies rather than accepting management narrative at face value.

The user prompt closes with nine numbered critical instructions including: "Distinguish FACTS (from filings) from INTERPRETATION (your analysis)"; "Compare QoQ and YoY trends"; "Rate earnings quality: accruals vs. cash flow alignment"; "Note all litigation or going-concern disclosures". The fact/interpretation separation mirrors sell-side research discipline and mitigates hallucination of financial figures not present in the excerpts.

Output includes `filing_quality_score` (0–10), `red_flags` list, `accounting_quality` section, and a final directional assessment (`filings-bullish` / `neutral` / `filings-bearish`).

For equity assets (TSLA), the filings agent passes the full `10k` and `10q` excerpts to the same Azure AI Foundry LLM, prompted as a forensic financial analyst. The required JSON output includes `overall_financial_health`, `margin_trajectory`, `balance_sheet_strength`, `management_credibility`, `overall_risk_profile`, and short/long-term investment horizon labels. When no filing text is present (BTC), the agent is skipped and its slot in the Final Answer context is set to `{"not_available": true}`.

4.7.5. Agent 5: Final Answer Synthesis (LLM)

Receives the complete JSON output of all four preceding agents and synthesises a single trading decision. The system prompt adopts the persona of **Howard Marks** (Oaktree Capital Management), enumerating his six core mental models: second-level thinking ("What does everyone else think, and why am I different?"), risk-before-return, cycle awareness, limits of knowledge, contrarian discipline, and "and then what?". Table 4 lists the hard constraints embedded in the prompt.

The prompt also requires explicit calculation of Expected Value $= p_{\text{win}} \cdot \text{gain} - p_{\text{loss}} \cdot \text{loss}$ and an approximate Kelly fraction, with a `reward_risk_ratio` computed inline before the decision is made. JSON fence stripping post-processing handles instruction-tuned models that wrap responses in markdown code blocks despite explicit instructions.

The final answer agent receives all three parallel agent outputs and the raw momentum label, then calls the Azure AI Foundry LLM to issue a definitive decision. The prompt encodes explicit confidence-to-decision rules (Listing 5) and requires a structured JSON response that includes a risk-reward decomposition alongside the recommended action.

Listing 5: Decision rules enforced in the Final Answer Agent prompt (`prompts.py`).

```
"""
RULES:
```

```
- NO BUY/SELL if reward_risk_ratio < 1.5 or
  expected_value_percent is negative
- confidence_thresholds:
    all_4_signals_aligned -> 80-95% (high_conviction)
    3_of_4_signals_aligned -> 65-80% (medium_conviction)
    2_of_4_signals_aligned -> 40-60% (HOLD_preferred)
    conflicting_signals    -> <40%   (HOLD_mandatory)
"""
```

The agent returns a JSON object from which `recommended_action` (BUY/HOLD/SELL) is extracted as the final system output. If the LLM returns invalid JSON, the pipeline defaults to HOLD.

4.8. Asset-Specific Adaptation

Because BTC lacks SEC filings, the filings agent is skipped entirely. Additionally, when the dataset's `history_price` vector is absent or empty for a given symbol, the pipeline synthesises a 20-point price series trending linearly up to the current price (Listing 6), ensuring the quantitative agent always has a valid input.

Listing 6: Synthetic price-history fallback when the dataset provides no history (`function_app.py`).

```
DEFAULT_HISTORY_POINTS = 20

def _build_history_series(price: float) -> list[dict]:
    history = []
    for i in range(DEFAULT_HISTORY_POINTS):
        # Trend linearly up to the current price
        step = float(price * (1 - 0.01 * (DEFAULT_HISTORY_POINTS - 1 - i)))
        history.append({
            "price":  step,
            "high":   step * 1.01,
            "low":    step * 0.99,
            "volume": 50_000_000,
        })
    return history
```

5. Prompt Engineering Framework

All three LLM agents share a common four-technique framework.

1. Persona grounding. Each system prompt opens with a named persona. Role grounding activates domain-specific reasoning patterns [9] and suppresses generic hedged language. The Howard Marks persona for Agent 5 introduces a risk-first cognitive frame, overriding the over-confident bullish bias observed in early tests without persona. Personas are described with enumerated capability domains (6 domains for Agents 3 and 4, 6 mental models for Agent 5), signalling the scope of expected output and preventing simple bullish/bearish labels in place of structured analysis.

2. Annotated JSON output schemas. Every output field carries an inline comment with its type and value range: `# -1.0 (very bearish) to +1.0 (very bullish)`. This allows token-by-token schema following during generation [10], reducing format violation rate from ∼12% (free-text format description) to ∼1.5% (inline annotations), observed across development runs.

3. Critical instruction repetition. Key instructions appear at both the start and end of the user prompt. The end uses ALL-CAPS phrasing: "Be SPECIFIC, cite actual content"; "QUANTIFY when

possible"; "Output VALID JSON only". This counteracts the lost-in-the-middle effect [11], where mid-prompt instructions are less reliably followed in long contexts. The combination of role declaration at the beginning and capitalised reinforcement at the end creates dual anchoring of the most critical constraints.

4. No-reasoning mode. All system prompts are prepended with:

```
Respond directly with JSON. Do not think step by step.
Do not use extended thinking. Do not explain your reasoning process.
```

Impact: average end-to-end latency dropped from 139 s to 62 s — a 55% reduction with no architecture changes. In qualitative review of 30 decision traces, removing extended thinking did not change any directional decision. The five-step chain-of-thought required in Agent 5's `reasoning` JSON field is produced *within* the structured response, not as a pre-response scratchpad.

6. Experimental Setup

6.1. Procedure

For each CSV row in the backtesting dataset, the system: (1) constructs the standard task payload; (2) submits via HTTP POST to the live Azure Function; (3) records the returned `decision`; (4) computes position P&L as $r_t = \text{sign}(a_t) \times (\Delta P_t / P_t)$ where $a_t \in \{+1, 0, -1\}$ for BUY/HOLD/SELL. No transaction costs, consistent with the competition protocol.

6.2. Ablation Configurations

We define seven ablation configurations to isolate the marginal contribution of each agent; the dedicated runs are left for future work (Section 7.4):

1. **Full:** All agents active.
2. **–Market Research:** Phase 1 skipped; quantitative agent uses the raw 20-point synthetic history.
3. **–Quantitative:** Agent 2 replaced with `not_available` stub.
4. **–News:** Agent 3 replaced with stub.
5. **–Filings:** Agent 4 replaced with stub (natural BTC control; identical to Full for BTC since filings are always absent).
6. **Quant Only:** Only Agent 2 active; Agents 3 and 4 stubbed.
7. **LLM Only:** Only Agents 3 and 4 active; Agent 2 stubbed.

Each ablation would substitute the removed agent with its graceful-degradation path, exactly mirroring the production behaviour when an agent times out or receives empty input.

7. Results on Test Dataset

7.1. Overall Performance

Among the measured baselines in Table 5, the Full Multi-Agent system is the only active strategy with positive cumulative return on both assets. Buy-and-Hold suffered severe drawdowns as both TSLA (-14.40%) and BTC (-27.69%) declined over this period, confirming that passive long exposure was the wrong posture. Momentum-only performed worst on TSLA (-34.67%, SR -1.65), because the momentum label is directionally correlated with recent price moves rather than future ones; for BTC it produced a milder loss (-6.45%) but with a drawdown twice that of our system. The Full Multi-Agent system reduced annualised volatility to 17.6% (TSLA) and 22.9% (BTC) versus 33.4% and 42.4% for Buy-and-Hold, confirming the Howard Marks reward/risk gate's role in capital preservation. This suggests that the system's risk controls reduced exposure during adverse periods.

Table 5
Overall system performance on the TSLA and BTC backtesting subset.. CR = Cumulative Return (%), SR = Sharpe Ratio, MD = Max Drawdown (%), DV = Daily Volatility (%), AV = Annualised Volatility (%). Metrics are from the backtesting run (2025-11-01 to 2026-04-30). The Full Multi-Agent numbers are backtested results from the Azure Function deployment.

	TSLA					BTC				
System	CR	SR	MD	DV	AV	CR	SR	MD	DV	AV
Buy-and-Hold	−14.40	−0.48	29.93	2.10	33.39	−27.69	−0.85	42.35	2.67	42.38
HOLD-only	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Momentum-only	−34.67	−1.65	46.61	2.06	32.70	−6.45	−0.01	30.06	2.67	42.44
Full Multi-Agent	+1.88	0.23	12.04	1.11	17.58	+2.46	0.26	13.48	1.44	22.89

Table 6
Distribution of the system's *final* decisions and the average next-day strategy return conditioned on that decision, computed from the BTC 181-day run (2025-11-01 to 2026-04-30; 36 BUY / 124 HOLD / 21 SELL days). These are final-action frequencies, *not* per-agent agreement rates; a full per-agent agreement analysis from individual signal logs is left for future work.

Final action	Days (%)	Avg r_{t+1} (%)
BUY	19.9	+0.09
SELL	11.6	+0.17[†]
HOLD	68.5	0.00

[†] Positive strategy return on SELL days indicates that executed short positions were profitable on average.

Table 7
Sample system decisions from the backtesting dataset. ΔP = `future_price_diff` ($). ✓ = matches theoretically optimal action; × = mismatch.

Asset	Date	ΔP	Decision	Match
TSLA	2025-08-01	+0.00	HOLD	✓
TSLA	2025-08-02	+0.00	HOLD	✓
BTC	2025-08-01	−558	SELL	✓
BTC	2025-11-05	−2,648	SELL	✓
BTC	2025-11-25	+3,004	BUY	✓
BTC	2025-11-30	−4,340	SELL	✓
TSLA	2025-11-16	+4.57	BUY	✓
TSLA	2025-11-05	−16.16	BUY	×

7.2. Final-Action Distribution and Conditional Returns

7.3. Sample Decisions from Backtesting

Table 7 shows a sample of system decisions from the backtesting dataset.

Key observations from the sample decisions:

- **TSLA Aug 1–2** (both `future_price_diff = 0.0`): Bearish momentum; heavily negative news (Florida Autopilot verdicts totalling $329M damages). System returns HOLD correctly: RSI is neutral, News Agent scores -0.65 but Howard Marks reward/risk gate (ratio < 1.5) overrides the potential SELL.
- **BTC Aug 1** ($\Delta P = -558$): Bearish momentum; negative news (crypto liquidations, leveraged-long blow-ups). System returns SELL correctly: bearish MACD, News sentiment -0.72, negative analyst consensus.
- **RSI = 100 edge case (TSLA row 42, 2025-09-12, $395.94):** Quantitative agent reported RSI = 100

Table 8
Agent-contribution ablation across the seven configurations of Section 6. Each row removes or isolates one agent via its `not_available` degradation path. "n/a" = BTC has no SEC filings, so the result is identical to Full. "—" = not evaluated in this work, left for future work.

Configuration	TSLA CR (%)	BTC CR (%)	Δ vs Full
Full (all agents)	+1.88	+2.46	0.00
–Market Research	—	—	—
–Quantitative	—	—	—
–News Analysis	—	—	—
–SEC Filings	—	n/a	—
Quantitative Only	—	—	—
LLM Only	—	—	—
Buy-and-Hold	−14.40	−27.69	—
HOLD-only	0.00	0.00	—

(valid when all 14 prior periods are up-days). Howard Marks persona correctly tempered this to HOLD, flagging the overbought extreme and insufficient news confirmation. Ground truth: `future_price_diff = 0.0`.

7.4. Ablation Study

We defined seven configurations in Section 6 to isolate each agent's marginal contribution, but the dedicated single-agent ablation runs were not completed for this submission and are left for future work. Table 8 therefore reports only the measured Full and baseline configurations; the per-agent rows are shown as "—". On the basis of the system's architecture we expect the following outcomes: (1) **–Market Research** is expected to produce the largest TSLA degradation, because the 30-day OHLCV window fetched in Phase 1 is the foundation of all 20+ indicator computations and the 20-point synthetic fallback degrades every indicator. (2) **–SEC Filings** has no effect on BTC (filings are always absent) but a measurable TSLA effect on earnings rows, where filings carry decisive forward-looking information. (3) **Quantitative Only** is expected to outperform **LLM Only** on BTC (no filings, noisier news) yet underperform on TSLA earnings days where filing forensics is decisive. (4) **Full Multi-Agent** is expected to dominate on risk-adjusted metrics (Sharpe, MD), because the Howard Marks reward/risk gate suppresses position-taking in ambiguous regimes.

7.5. Cross-Asset Generalisation

Table 9 reports Full Multi-Agent performance across all 12 assets in the extended backtesting suite, using an identical system configuration and the same 181-day evaluation window (2025-11-01 to 2026-04-30). The two official competition assets (TSLA, BTC) are shown in bold for reference.

The results reveal two distinct performance profiles. The system performs well on assets exhibiting clear directional trends supported by aligned news and fundamental signals: MRNA (+40.76%, SR 1.76) and GOOGL (+24.48%, SR 1.48) benefited from high-conviction earnings-driven moves where the quantitative, news, and filings agents concurred, consistent with the hypothesis in Section 7.5 that filing forensics provides decisive signal on earnings rows. Across the ten equities, six of ten produce positive returns, suggesting the system generalises beyond the two official competition assets.

Assets in sustained downtrends — ADBE, MSFT, and ETH — produce negative returns because the Howard Marks reward/risk gate (Section 4.7.5) suppresses short positions when the reward-to-risk ratio falls below 1.5, causing the system to hold rather than sell into protracted bear moves. ETH's underperformance (−35.25%) contrasts sharply with BTC (+2.46%): despite identical filings-absent routing, ETH's higher annualised volatility (35.1% vs. 22.9%) and noisier news signal push Agent 5 toward HOLD even during unambiguous downtrends, confirming that graceful degradation increases HOLD bias under high uncertainty (Section 3).

Table 9
Cross-asset performance of the Full Multi-Agent system. CR = Cumulative Return (%), SR = Sharpe Ratio, MD = Max Drawdown (%), AV = Annualised Volatility (%), WR = Win Rate (%). Assets sorted by CR within each class. **Bold** = official competition assets.

Symbol	Class	CR	SR	MD	AV	WR
Equities						
MRNA	Equity	+40.76	1.76	9.13	29.33	40.0
GOOGL	Equity	+24.48	1.48	16.39	22.17	37.3
AMZN	Equity	+8.78	0.81	13.35	16.02	40.5
AAPL	Equity	+8.35	0.88	8.98	13.84	36.4
META	Equity	+7.40	0.53	21.64	24.43	31.3
TSLA	Equity	+1.88	0.23	12.04	17.58	34.9
NVDA	Equity	−5.97	−0.26	20.90	22.93	36.3
BMRN	Equity	−7.14	−1.41	11.19	7.12	16.7
MSFT	Equity	−17.86	−1.21	23.22	20.83	32.1
ADBE	Equity	−20.83	−1.47	28.86	20.62	35.0
Cryptocurrencies						
BTC	Crypto	+2.46	0.26	13.48	22.89	57.9
ETH	Crypto	−35.25	−1.54	38.63	35.07	51.5

8. Limitations

Point-in-time data integrity. The Phase 1 market-research enrichment issues live Finnhub calls (real-time quote, 30-day OHLCV window, analyst recommendations, financial metrics, and news) at execution time rather than reconstructing each field strictly as of the historical evaluation date. For genuinely historical series — notably the 30-day OHLCV window ending on the evaluation date — Finnhub returns point-in-time values and no look-ahead is introduced. However, snapshot fields such as the current analyst consensus and the latest financial metrics reflect their state at run time and may embed information that post-dates the evaluation day, introducing potential look-ahead bias. We did not fully isolate point-in-time snapshots for these fields in the reported runs; the results should therefore be read as an optimistic bound on strictly causal live performance. Crucially, the deterministic quantitative agent, which contributes the largest single share of the composite score (Section 4.7.5), operates only on price history up to and including the evaluation date and is unaffected. A fully point-in-time pipeline that pins every Finnhub field to the evaluation date is left for future work.

Per-agent ablations left to future work. Only the full-system and baseline configurations in Table 8 are measured; the dedicated single-agent ablation runs were not completed for this submission. The per-agent interpretation in Section 7.4 should therefore be read as architecture-driven expectation rather than confirmed measurement.

Finnhub headlines-only inputs. The news signal is built from Finnhub headlines and short summaries rather than full article bodies (Section 3), trading a small amount of context for deterministic latency. This may under-represent details buried in article bodies.

LLM hallucination on filings. The SEC filings agent receives only excerpts, not complete filings. The fact/interpretation separation in the prompt (Section 4.7.4) mitigates, but cannot fully eliminate, the risk of the model synthesising figures not present in the provided text.

9. Conclusion

We presented a five-agent hybrid system for FinMMEval Task 3. The central principle is the complementarity of deterministic quantitative signals and LLM qualitative reasoning: technical indicators are precise, reproducible, and fast; LLMs process regulatory filings and interpret management tone in ways that rule-based systems cannot.

Three engineering decisions proved decisive: (1) Phase 1 market enrichment ensures the quantitative anchor operates on real rather than synthetic data; (2) parallel Phase 2 execution keeps the orchestration within the 3-minute budget; (3) no-reasoning mode eliminated 55% of average latency with no measurable quality loss.

A per-agent ablation is defined but left to future work (Sections 7.4, 8); on the basis of the architecture we expect the Full system to dominate on risk-adjusted metrics, with the strongest hybrid advantage on TSLA earnings days where qualitative filing signals provide information the quantitative layer cannot access.

Acknowledgments

This work was conducted as part of Shell's participation in the FinMMEval Lab at CLEF 2026. The authors thank the FinMMEval organisers at MBZUAI for the backtesting datasets and evaluation infrastructure.

Declaration on Generative AI

During the preparation of this work, the authors used Claude Sonnet 4.6 and ChatGPT/GPT-5.5 Pro in order to assist with drafting and editing of LaTeX source and grammar and spelling checks. After using this tool, the authors reviewed and edited all AI-assisted content as needed and take full responsibility for the publication's content.

References

[1] Z. Xie, Y. Dai, R. Elbadry, V. Jani, X. Peng, L. Qian, G. Georgiev, D. Dimitrov, F. Zhang, J. Huang, J. Geng, Y. Chen, Y. Yuan, H. Wu, Y. Wang, I. Koychev, V. Stoyanov, M. Song, Y. Chen, S. Liu, P. Nakov, Overview of FinMMEval 2026: Multilingual and multimodal financial evaluation, in: Experimental IR Meets Multilinguality, Multimodality, and Interaction. Proceedings of the Seventeenth International Conference of the CLEF Association (CLEF 2026), Lecture Notes in Computer Science (LNCS), Springer, Jena, Germany, 2026. September 21–24.

[2] Z. Xie, L. Qian, G. Georgiev, D. Dimitrov, R. Elbadry, F. Zhang, X. Peng, J. Huang, V. Jani, Y. Dai, J. Geng, Y. Chen, Y. Yuan, H. Wu, Y. Wang, I. Koychev, V. Stoyanov, M. Song, Y. Chen, S. Liu, P. Nakov, Overview of the FinMMEval 2026 task 3: Financial decision making, in: CLEF 2026 Working Notes, CEUR Workshop Proceedings, CEUR-WS.org, Jena, Germany, 2026. September 21–24.

[3] J. J. Murphy, Technical Analysis of the Financial Markets, New York Institute of Finance, New York, NY, 1999.

[4] H. Yang, X.-Y. Liu, C. D. Wang, FinGPT: Open-source financial large language models, arXiv:2306.06031, 2023. arXiv:2306.06031.

[5] S. Wu, et al., BloombergGPT: A large language model for finance, arXiv:2303.17564, 2023. arXiv:2303.17564.

[6] H. Yang, et al., FinRobot: An open-source AI agent platform for financial applications, arXiv:2405.14767, 2024. arXiv:2405.14767.

[7] Y. Li, S. Wang, H. Ding, H. Chen, Large language models in finance: A survey, in: Proceedings of ICAIF 2023, 2023, pp. 374–382.

[8] L. Qian, X. Peng, Y. Wang, V. J. Zhang, H. He, H. Smith, Y. Han, Y. He, H. Li, Y. Cao, Y. Yu, A. Lopez-Lira, P. Lu, J.-Y. Nie, G. Xiong, J. Huang, S. Ananiadou, When agents trade: Live multi-market trading benchmark for LLM agents, arXiv:2510.11695, 2025. URL: https://arxiv.org/abs/2510.11695. doi:10.48550/arXiv.2510.11695. arXiv:2510.11695.

[9] J. Wei, et al., Chain-of-thought prompting elicits reasoning in large language models, in: Advances in Neural Information Processing Systems (NeurIPS), volume 35, 2022, pp. 24824–24837.

[10] Y. Zhou, et al., Large language models are human-level prompt engineers, in: International Conference on Learning Representations (ICLR 2023), 2023. arXiv:2211.01910.

[11] N. F. Liu, et al., Lost in the middle: How language models use long contexts, Transactions of the Association for Computational Linguistics (TACL) 12 (2024) 157–173.