

Toward Expert Investment Teams: A Multi-Agent LLM System with Fine-Grained Trading Tasks

Kunihiro Miyazaki
 Japan Digital Design, Inc.
 Tokyo, Japan
 kunihirom@acm.org

Stephen Roberts
 Department of Engineering Science
 University of Oxford
 United Kingdom
 sjrob@robots.ox.ac.uk

Takanobu Kawahara
 Japan Digital Design, Inc.
 Tokyo, Japan
 takanobu.kawahara@japan-d2.com

Stefan Zohren
 Department of Engineering Science
 University of Oxford
 United Kingdom
 stefan.zohren@eng.ox.ac.uk

Abstract

The advancement of large language models (LLMs) has accelerated the development of autonomous financial trading systems. While mainstream approaches deploy multi-agent systems mimicking analyst and manager roles, they often rely on abstract instructions that overlook the intricacies of real-world workflows, which can lead to degraded inference performance and less transparent decision-making. Therefore, we propose a multi-agent LLM trading framework that explicitly decomposes investment analysis into fine-grained tasks, rather than providing coarse-grained instructions. We evaluate the proposed framework using Japanese stock data, including prices, financial statements, news, and macro information, under a leakage-controlled backtesting setting. Experimental results show that fine-grained task decomposition significantly improves risk-adjusted returns compared to conventional coarse-grained designs. Crucially, further analysis of intermediate agent outputs suggests that alignment between analytical outputs and downstream decision preferences is a critical driver of system performance. Moreover, we conduct standard portfolio optimization, exploiting low correlation with the stock index and the variance of each system's output. This approach achieves superior performance. These findings contribute to the design of agent structure and task configuration when applying LLM agents to trading systems in practical settings.

CCS Concepts

- Applied computing → Economics; Business-IT alignment;
- Computing methodologies → Multi-agent systems; Natural language processing.

Keywords

LLM, Multi-Agent System, Trading, Investing, Prompt Design

1 Introduction

The rapid advancement of Large Language Models (LLMs) has led to high expectations that they function as an autonomous workforce, like human employees, in practical domains [36, 50]. Consequently, many companies have initiated the internal and external implementation of “AI agents” [4, 46, 53]. In the financial industry, the development of AI agents has begun across various applications [9, 39],

with expectations to generate profits in both investment and trading contexts through the utilization of agent-based autonomy and intelligence [34, 62]. One of the most common configurations is a multi-agent trading system, in which multiple LLM agents are assigned specific roles such as fundamental analysis, news parsing, and risk management [62, 68, 72].

However, considering practical applications, concerns exist that current LLM trading systems are constructed with a simplified task design, given the complexity of investment analyst tasks. Most existing studies adopt coarse-grained task settings, primarily assigning roles and high-level objectives to agents. For instance, a fundamental agent tends to be simply instructed to “analyze financial statements (e.g., 10-K)”, leaving unexplored the fine-grained task design for both qualitative and quantitative analyses typically performed in real-world situations [62, 68, 72].

Providing coarse-grained instructions to LLMs for complex tasks presents two major challenges. The first is performance degradation. It has been reported that overly vague instructions can reduce the output quality of LLMs [21, 74], and when tasks are too complex, LLMs have been observed to occasionally cease reasoning midway or abandon reasoning entirely [52, 60]. The second is the lack of interpretability. When LLMs are given ambiguous instructions, typically only the final output is visualized, making it impossible to interpret the intermediate reasoning process [43, 71]. In such cases, practical deployment becomes difficult, especially in asset management practices where significant capital is at stake [18].

To address these issues, this study constructs a multi-agent LLM trading system that assigns detailed, concrete investment decision-making tasks to trading agents, based on real-world practices of investment analysts. In contexts beyond finance, it has been reported that providing expert processes to LLM agent systems is effective [14]. Furthermore, separating *task planning* from *execution* is reported to be effective in LLM agent systems [49, 57]. It is also noted that performance improves when domain experts design the tasks [28]. Inspired by these studies, we anticipate that tracing the complex workflows of real-world investment analysts will enhance the performance of investment agents. Furthermore, responding to concrete tasks is expected to improve the agents’ output explainability [13, 22, 66], which is crucial for industrial applications [18].

In the experiment, we conduct backtesting to evaluate whether our proposed fine-grained task configuration leads to performance

improvements. The evaluation uses Japanese equity market data, including stock prices, financial statements, news articles, and macroeconomic information. To prevent data leakage and account for the LLM model’s knowledge cutoff, we set the backtesting period from September 2023 to November 2025. Beyond portfolio-level performance, we also evaluate intermediate textual outputs to analyze how task decomposition affects reasoning behavior and interpretability. Finally, to demonstrate real-world applicability, we verify the strategy’s effectiveness through portfolio optimization benchmarked against market indices.

This study makes the following contributions. **Impact of Task Granularity:** We focus on task design for LLM-based trading agents, which has been largely overlooked in prior works, and demonstrate through controlled experiments that fine-grained task decomposition improves agent performance. **Agent Ablation Analysis:** We conduct a comprehensive ablation study by systematically removing and replacing individual agents, providing new insights into the functional roles of agents in multi-agent trading systems. **Real-world Evaluation:** We emphasize real-world applicability by adopting realistic problem settings (e.g., agent roles and team architecture) and evaluating not only portfolio-level performance but also intermediate textual outputs and portfolio optimization results benchmarked against market indices. **Reproducibility:** To support reproducibility and future research, we release the implementation codes with prompts upon acceptance.

2 Related Work

2.1 Multi-Agent Trading Systems with LLMs

In the emergence of LLM trading systems, whereas early studies primarily adopted single-agent architectures [34], more recent work has shifted toward multi-agent trading systems that more closely resemble real investment teams [51]. In such systems, multiple agents are assigned complementary roles and collaborate to process heterogeneous financial information [e.g. 24, 62, 64, 68, 72].

The focus of existing research on multi-agent LLM trading can broadly be grouped into two categories [51]. The first category involves the efforts concerning their organizational structure and roles, such as agents’ arrangements, interactions between agents, and role diversification, often inspired by real financial institutions. These typically adopt a manager–analyst architecture, where a manager coordinates several specialized analyst agents that collect, filter, and synthesize information from various sources [68]. Analyst roles span a wide range of functions, including fundamental analysis [72], technical analysis [64], news sentiment extraction [62], and risk management [29, 67]. The second category focuses on reinforcement learning-based approaches, which aim at improving decision policies through iterative feedback. Notable directions include Reflection mechanisms, which incorporate realized trading outcomes into subsequent reasoning, and Layered Memory architectures that regulate the temporal scope of accessible information [55, 68, 70].

Our research builds on the first research focus, namely, structure and roles. While prior studies carefully design both the structure and roles of agents, the prompts that specify how each role should operate have not been fully explored—they are often defined at a relatively coarse level, without explicitly aligning them with real investment tasks. This is presumably because the field of multi-agent

systems for LLM-based trading is nascent; research has primarily focused on achieving end-to-end trading completion, with little light shed on the granular details of task execution. In contrast, we aim to formulate role prompts in a more fine-grained and realistic manner that deliberately imitates the division of labor in real investment organizations, to improve controllability and interpretability of trading performance. Furthermore, despite the growing interest in agent-based LLM trading systems, we believe that agent-level ablation studies remain relatively underexplored; thus, by conducting a systematic analysis, we provide important practical insights.

2.2 Prompt Designs and Expert Task Settings

Recent LLM research explores whether explicitly planning and decomposing problems (rather than providing simple and ambiguous instructions) can improve performance on complex tasks. Frameworks such as MetaGPT [14] and Agent-S [1] demonstrate that encoding Standard Operating Procedures (SOPs) into multi-agent systems can reduce errors and enhance output quality, particularly in software-engineering tasks. Related approaches, including the “plan-and-execute” [57] and “blueprint-first” [49] paradigms, further suggest that fixing workflow structures—rather than allowing LLMs to autonomously determine task sequences—helps stabilize long-context reasoning. Taken together, these results indicate that embedding expert knowledge into prompts can improve reliability and performance, consistent with findings reported in Li et al. [28].

In contrast, the financial domain is only beginning to explore how such “expert processes” can be formalized for LLMs. Existing approaches often rely on agents inspired by real or virtual investor personas [7, 67] or on systems that allow humans to intervene during reasoning [58]. However, these systems typically do not treat expert workflows themselves as explicit structural components.

The work most closely related to ours is Financial Chain-of-Thought (CoT) prompting introduced in FinRobot [20, 65]. By pre-structuring financial analysis into predefined sections, FinRobot encourages domain-specific reasoning when generating analysis reports. Our approach differs from the prior work in several respects. First and foremost, we formalize expert workflows as fixed analysis protocols, rather than relying on generic CoT-style prompting. Second, rather than focusing on report generation, we aim to operationalize trading decisions. Third, we extend beyond firm-level analysis by incorporating other factors such as macroeconomic information and sector information.

3 Problem Setting

The primary aim of this study is to investigate whether providing fine-grained task specifications to LLM agents in automated trading contributes to improved operational performance. To validate the effectiveness of the multi-agent system with the proposed setting, we conduct evaluations under constraints that mimic the investment practices of institutional investors. Specifically, we construct and backtest a monthly rebalancing portfolio based on a long-short strategy targeting large-cap stocks in Japanese equity markets.

3.1 Backtesting and Model Setup

Investment Universe: We use the TOPIX 100 constituents, representing stocks with the highest market capitalization in Japan.

Portfolio Construction: To eliminate market-wide volatility risk and isolate stock selection capability, we adopt a market-neutral strategy. Specifically, the portfolio holds an equal number of stocks for both long (buy) and short (sell) positions with equal weighting.

Rebalancing Frequency: We conduct portfolio rebalancing at the opening of the first business day of each month.

Test Period: The evaluation covers the period from September 2023 to November 2025, totaling 27 months.

Model Selection and Look-ahead Bias Mitigation: We employ state-of-the-art LLMs, specifically the GPT-4o [15] with the knowledge cutoff date in August 2023, as an inference model. Previous research warns that LLMs may “memorize” historical financial time-series data present in their training corpora, posing a risk of look-ahead bias that artificially inflates backtesting results [35]. To guarantee the validity of our backtest, it is crucial to strictly separate the agent’s knowledge from future events [10, 16]. In this experiment, we enforce strict temporal ordering by feeding the agents only text and market data publicly available up to the specific decision point and by conducting the backtesting period after the LLM’s knowledge cutoff date, thereby preventing information leakage. For the model setting, we use the default, such that the temperature is set to unity. While a temperature of zero might seem preferable for reducing variation, it does not fully eliminate stochasticity [3, 42]. Since we aggregate multiple outputs using the median in our experiment, variability is already mitigated. Also, from an ensemble perspective, a temperature of unity can be preferable as it preserves useful output diversity [8, 59]. As for the inference time, we note that since we conduct a monthly rebalancing, real-time processing is unnecessary.

3.2 Evaluation Metrics

We assess the performance of the agent-generated portfolios using both quantitative and qualitative metrics.

Quantitative Metrics: We use the standard measure of risk-adjusted return, namely the Sharpe ratio. We calculate it as the mean of monthly portfolio returns divided by their standard deviation.

Qualitative Metrics: We analyze the scores and reasoning texts generated by the analyst agents to assess how these agents communicate with the manager agents.

4 Multi-Agent Framework Configuration

Prior to detailing the fine-grained tasks assigned to each agent, this section outlines our baseline configuration of the LLM investment team. While multi-agent systems generally allow for flexible composition of agents [e.g., 55, 62, 64, 68, 72], from a practical industry perspective, we aim to assemble the realistic components emulating the operational workflows of professional institutional investors.

4.1 Hierarchical Decision-Making Process

The system adopts a bottom-up manager-analyst framework for decision-making, where information is progressively abstracted and aggregated through a multi-level hierarchical structure from analysts to the portfolio manager. Figure 1 depicts an overview of our trading system composition.

Level 1. Information Aggregation and Scoring (Analyst Agents): Four types of specialist agents—Quantitative, Qualitative, News,

and Technical—analyze each stock in the investment universe. The Quantitative and Technical Agents assign attractive scores $S \in [0, 100]$ and generate a textual rationale supporting their evaluation based on their specific domain expertise. The Qualitative and News Agents generate the supplemental information (scores and texts). The output scores and reports are integrated and submitted to the Sector Agent.

Level 2. Sector-Level Adjustment (Sector Agent): The Sector Agent aggregates scores and reports from subordinate analysts and adjusts them based on sector-specific benchmarks. Specifically, a stock’s quantitative valuation score is re-evaluated against sector averages. The adjusted scores and reports are submitted to the Portfolio Manager (PM) agent.

Level 2. Macro-Environmental Assessment (Macro Agent): Independently, the Macro Agent analyzes broader economic indicators, such as interest rate trends, business cycles, and foreign exchange dynamics. It evaluates the current market regime and submits scores and texts to the PM Agent.

Level 3. Final Portfolio Construction (PM Agent): The PM Agent synthesizes the scores and reports from the Sector Agent and the Macro Agent to determine final scores for all stocks in the TOPIX 100. Then, we select the same number of stocks for long positions with the highest scores and the short positions with the lowest scores to construct a portfolio.

4.2 Data Sources

The agents access a combination of structured and unstructured data to inform their decisions. Considering reproducibility, we adhere to open data as much as possible.

Stock Price Data: We use daily prices of stocks in TOPIX 100 obtained from Yahoo Finance [2]. We use close prices to calculate the metrics for investment decisions, and use open prices for execution and performance evaluation. To clarify, we conduct rebalancing after the market close of the last business day of a month and execute it at the open of the first business day of the following month.

Financial Statements Data: We utilize quarterly, semi-annual, and annual securities reports of companies. We retrieve these documents through the EDINET API [11] managed by the Financial Services Agency (FSA) of Japan. We use both numerical financial statements and qualitative textual information.

News Data: To capture sentiment and events (e.g., scandals, product launches) related to companies, we aggregate headlines and freely accessible article previews from major financial news outlets, including *Nikkei*, *Reuters*, and *Bloomberg* (all Japanese editions) sourced from Ceeek.jp News [6], a comprehensive aggregator of virtually all Japanese news media articles.

Macroeconomic Data: To gauge the macroeconomic environment, we compile a comprehensive dataset using the FRED [12] and Yahoo Finance. We use the metrics from four dimensions: **Rates & Policy** includes the US Federal Funds Rate, US 10Y Treasury Yield, JP Policy Rate, and JP 10Y JGB Yield; **Inflation & Commodities** covers US and JP CPI, along with Gold and Crude Oil prices; **Growth & Economy** tracks US Non-Farm Payrolls, Industrial Production, Housing Starts, Unemployment Rate, and the JP Business Conditions Index; and **Market & Risk** indicators include the USD/JPY exchange rate, major equity indices (Nikkei 225, S&P 500), and volatility indices

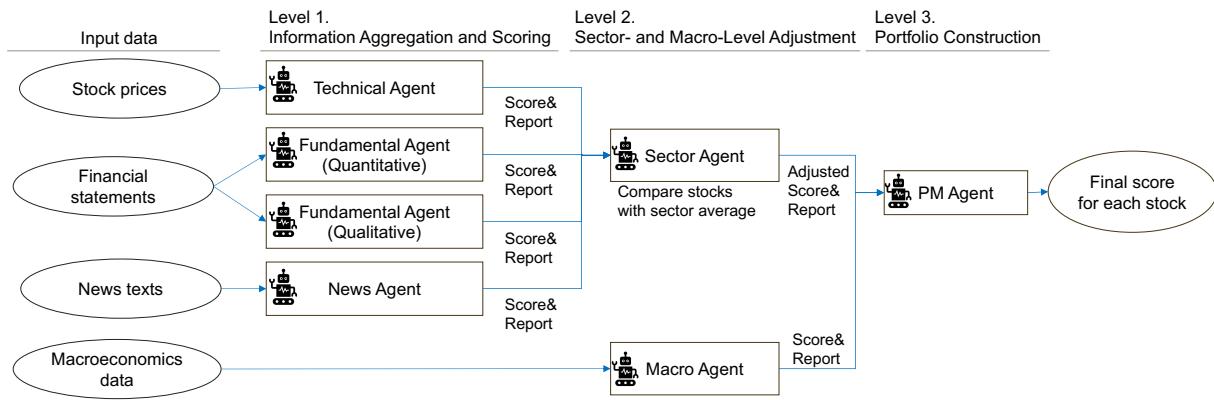


Figure 1: Overview of our multi-agent LLM trading system (see main text for details).

(US VIX, Nikkei VI). For all indicators, we utilize the latest available closing values and their month-over-month rates of change.

5 Methodology

This section details the fine-grained tasks we provide to agents in prompts and the design of experiments to verify their effectiveness.

The core methodological contribution of this study lies in the granularity of instructions (prompts) for AI agents. Theoretically, the decomposition of tasks can be arbitrarily granular. However, to operationalize automated trading, we define fine-grained tasks as standard analytical tasks that should be performed routinely in practical operations by professional analysts. While these tasks are fundamental to human practitioners, they have received limited attention in existing financial multi-agent research.

5.1 Tasks and Prompt Design

Here, we define the tasks for the seven individual agents. Among these, the Technical and Quantitative Agents are the targets of our experiments; thus, we describe both fine-grained and coarse-grained tasks for these agents. We selected these two agents because, among the other agents, only they produce outputs that directly lead to investment execution and their numerical data processing can be clearly defined. We provide the full prompts in Appendix B.

1. Technical Agent (Technical Analysis). This agent analyzes stock price movement to generate attractive scores ranging from 0 to 100 for each stock.

For **fine-grained tasks**, we pass pre-calculated technical indicators commonly used in market analysis to the Technical Agent. To eliminate the bias of nominal price levels, all indicators are normalized or ratio-adjusted. We refer existing research on the selection of indicators [e.g., 30, 32, 38, 44, 61, 62, 64, 70].

Momentum: Rate of Change (RoC): We compute the RoC across multiple lookback horizons (5, 10, 20, and 30 days; 1, 3, 6, and 12 months) to capture trend strength and temporal persistence.

Volatility: Bollinger Band: Instead of price bands, we identify statistically extreme levels using the Z-score formula $Z = (P - \mu_{20})/\sigma_{20}$, which normalizes the price deviation from the 20-day moving average by the standard deviation.

Oscillator: Moving Average Convergence Divergence (MACD): This metric measures trend momentum as the difference between short-term (12-day) and long-term (26-day) exponential moving averages (EMA) of price, with a 9-day EMA of the MACD line as the “signal line,” and their difference as the “histogram.” We show the detailed formulations in Appendix A.

Oscillator: Relative Strength Index (RSI): This metric compares the magnitude of recent gains and losses over a 14-day lookback period to identify potentially overbought or oversold conditions. We show the detailed formulations in Appendix A.

Oscillator: (KDJ): We calculate the stochastic oscillator $\%K$ (price position within the 9-day high-low range), $\%D$ (3-day moving average of $\%K$), and the divergence $J = 3D - 2K$ to capture potential trend reversals. We show the detailed formulations in Appendix A.

For **coarse-grained tasks** for the Technical Agent, instead of adding pre-calculated metrics, we add the raw data directly to the prompts. Specifically, we feed the data used to calculate the metrics employed in fine-grained tasks (i.e., daily prices over one year) directly into the agent.

2. Quantitative Agent (Quantitative Fundamentals). This agent quantitatively evaluates a company’s financial health and growth potential based on the numbers on financial statements, then generates attractive scores ranging from 0 to 100 for each stock.

For all metrics, we provide the agent with the numerical values and the RoC compared to the previous year, regardless of the granularity of tasks. For flow variables (e.g., ROE and Sales), we calculate valuation metrics using Trailing Twelve-Month (TTM) to mitigate seasonality while incorporating the most recent performance [47]. For stock variables (e.g., Equity Ratio, Total Assets), we utilize values from the latest quarterly balance sheet to ensure timeliness [33]. When a metric cannot be computed due to insufficient data, we pass NaN values to the LLM and rely on the model to handle missing data appropriately.

For **fine-grained tasks**, as in the Technical Agent, we follow the standard practices in financial analysis [e.g., 25, 31, 48] to calculate the traditional investment metrics with the five dimensions: **Profitability:** ROE, ROA, Operating Profit Margin, FCF Margin; **Safety:** Equity Ratio, Current Ratio, D/E Ratio; **Valuation:** P/E

Ratio, EV/EBITDA Multiple, Dividend Yield; **Efficiency**: Total Asset Turnover, Inventory Turnover Period; and **Growth**: Revenue Growth Rate (CAGR), EPS Growth Rate.

For **coarse-grained tasks** for the Quantitative Agent, instead of providing aggregated financial metrics, we provide the agent with raw data points that we can get from financial statements, such as: **Income Statement**: Sales, Operating Profit, Net Income, Cost of Sales, Depreciation; **Balance Sheet**: Total Assets, Equity, Cash, Receivables, Financial Assets, Inventory, Current Liabilities, Interest Bearing Debt; **Cash Flow**: Operating, Investing; and **Market Data**: Monthly Close, EPS, Dividends, Issued Shares. Additionally, historical EPS data (1-year and 3-year lookbacks) are included to assess long-term earnings stability.

3. Qualitative Agent (Qualitative Fundamentals). This agent analyzes unstructured text data from securities reports to evaluate sustainability and competitive advantages that are not captured by numerical metrics. It extracts information from specific sections and analyzes it based on predetermined evaluation criteria: **Business Overview** evaluates business model robustness based on “History,” “Business Description,” and “Affiliated Companies;” **Risk Analysis** extracts potential downside risks from “Business Risks” and “Issues to Address;” **Management Policy** evaluates strategic intent and execution capability based on “Management’s Discussion and Analysis (MD&A);” **Governance** assesses management transparency via board composition (proportion of outside directors) and “Corporate Governance Status.” The agent outputs business, risk, and management scores (5-point scale) and reasoning texts as input to the Sector Agent.

4. News Agent (News Sentiment and Events). This agent aggregates recent news headlines and summaries from major economic media outlets. It detects material events such as earnings revisions, scandals, new product announcements, and M&A activities. For each company, the system inputs news data from the current month; if no news is available, we use NaN as input. The agent searches for headlines containing company names (and their abbreviations) and extracts the headline and content when found. The agent outputs risk and return outlook scores (5-point scale) and reasoning texts as input to the Sector Agent.

5. Sector Agent (Sector-Level Adjustment). This agent synthesizes the outputs from the four analyst agents and compares the quantitative figures of each stock with the sector averages. The agent provides the re-evaluated attractive score (0-100 scale) and an investment thesis to the PM agent.

6. Macro Agent (Macro-Environmental Assessment). This agent analyzes the economic environment with five dimensions—Market Direction, Risk Sentiment, Economic Growth, Interest Rates, and Inflation—based on the absolute levels and month-to-month changes of JP/US economic indicators, then provides the scores (0-100 scale) for each dimension and reasoning texts to the PM agent.

7. PM Agent (Final Portfolio Construction). This agent integrates the bottom-up view (from the Sector Agent) with the top-down view (from the Macro Agent), then generates a final attractive score (0-100 scale) for the long-short portfolio construction.

6 Experimental Results

In this section, we present the empirical backtesting results of the proposed multi-agent trading system using the Japanese TOPIX 100 universe from September 2023 to November 2025.

6.1 Fine-Grained vs Coarse-Grained Tasks

To evaluate the effectiveness of fine-grained task decomposition, which is our main objective, we compare the performance of the proposed method (agents with fine-grained tasks) against the baseline method (with coarse-grained tasks) across varying portfolio sizes ($N \in \{10, 20, 30, 40, 50\}$). Note that $N = 10$, for instance, indicates that we long five stocks and short five stocks. We conduct the experiments in two settings: using all agents and agents without one agent (leave-one-out).

6.1.1 Comparison using All Agents. Figure 2 shows the comparison of Sharpe ratios between fine-grained (pink) and coarse-grained (blue) task settings across five different portfolio sizes, with each configuration evaluated over 50 independent trials. Stars (*) indicate Mann-Whitney U test significance: $p < 0.0001, 0.001, 0.05$ shown as ****, ***, *; ‘ns’ indicates not significant. Hereafter, we use the same notation throughout the paper. As indicated by the stars, the agents with the fine-grained tasks significantly outperform their coarse-grained counterparts in 4 out of the 5 tested horizons (20, 30, 40, and 50 stocks). The only exception is the initial case (10 stocks), denoted as “ns” (not significant), which may be attributed to the relatively small number of stocks that renders the backtesting results noisy and unstable. Overall, despite the noise in the smallest setting, the aggregate trend demonstrates that providing detailed (fine-grained) information contributes to superior risk-adjusted returns.

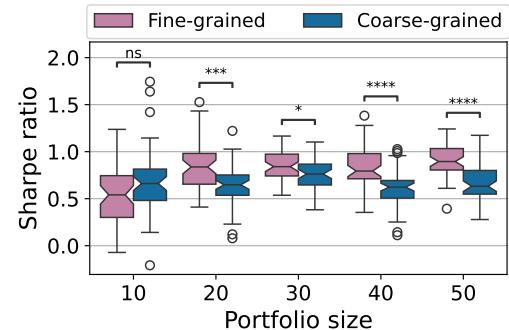


Figure 2: Sharpe ratios for fine-grained (pink) and coarse-grained (blue) settings across portfolio sizes. Box plot notches represent the 95% confidence interval of the median.

6.1.2 Comparison with Leave-One-Out Settings. We further compare fine-grained and coarse-grained agent settings under leave-one-out settings. Specifically, we systematically remove one of the bottom-level specialist agents—namely, Technical, Quantitative, Qualitative, News, and Macro—to assess the robustness of the proposed method’s superiority. Table 1 reports the differences in median Sharpe ratio between fine-grained and coarse-grained settings, as $\Delta SR = \text{Median}(SR_{\text{fine}}) - \text{Median}(SR_{\text{coarse}})$. The top row (“All agents”)

corresponds to the all-agent configuration, consistent with the results shown in Figure 2. Across leave-one-out settings, we find

Table 1: Sharpe Ratio Differences Between Fine-grained and Coarse-grained Settings

Portfolio size	10	20	30	40	50
All agents	-0.12	+0.19****	+0.08*	+0.17****	+0.26****
w/o Technical	+0.54***	-0.07	-0.34***	-0.66****	-0.79****
w/o Quant.	+0.1	+0.04	+0.16	+0.2*	+0.12
w/o Qual.	+0.49***	+0.24*	+0.41***	+0.55****	+0.33***
w/o News	+0.35*	+1.0***	+1.04***	+1.04***	+1.08***
w/o Macro	+0.11	+0.1	+0.23	+0.31*	+0.01

that the differences are predominantly positive in most configurations. This indicates that the fine-grained architecture generally achieves higher Sharpe ratios than the coarse-grained baseline, even when specific analytical perspectives are removed. A notable exception is the “w/o Technical” setting, where the performance reverses for larger portfolio sizes, suggesting that the Technical Agent plays a central role in driving the performance advantage of fine-grained task decomposition. Overall, the results demonstrate that the fine-grained task design robustly outperforms the equivalent coarse-grained design in backtesting.

6.2 Ablation Studies

We conduct ablation studies to quantify the contribution of each specialized agent to overall performance. Table 2 presents the changes in the Sharpe ratios compared to the “All agents” baseline. The values represent the difference calculated as $\text{SR}_{\text{ablation}} - \text{SR}_{\text{baseline}}$. Consequently, a positive value indicates that removing the agent improved performance (implying the agent was detrimental or noisy), while a negative value indicates that performance degraded (implying the agent was beneficial). Overall, most ablation settings show

Table 2: Trading performance under Ablation Settings

(a) Fine-grained settings					
	10	20	30	40	50
All agents (Baseline)	0.54	0.84	0.84	0.79	0.9
w/o Technical	+0.13	-0.42****	-0.4****	-0.56****	-0.66****
w/o Quant.	+0.45****	+0.23***	+0.41****	+0.48****	+0.2****
w/o Qual.	+0.16	+0.27***	+0.47****	+0.5****	+0.33****
w/o News	+0.21	+0.23**	+0.29***	+0.12	+0.25****
w/o Macro	+0.28*	+0.15*	+0.27*	+0.36****	+0.16***

(b) Coarse-grained settings					
	10	20	30	40	50
All agents (Baseline)	0.66	0.65	0.76	0.62	0.63
w/o Technical	-0.52****	-0.16*	+0.02	+0.28***	+0.4****
w/o Quant.	+0.24***	+0.38****	+0.33****	+0.46****	+0.34****
w/o Qual.	-0.45****	+0.22***	+0.14	+0.13	+0.26****
w/o News	-0.26**	-0.59****	-0.68****	-0.75****	-0.57****
w/o Macro	+0.05	+0.23****	+0.11	+0.22*	+0.41***

positive differences relative to the full-agent configuration, indicating that most individual agents may introduce noise or redundant

signals. This highlights the importance of carefully designing agent roles and interactions, rather than simply increasing the number of specialized components.

In the fine-grained setting (Table 2a), however, the “w/o Technical” condition shows predominantly negative differences, especially for larger portfolio sizes. This implies that the Technical Agent provides particularly strong predictive signals. In contrast, removing other agents (Macro, Quantitative, Qualitative) often results in positive differences, suggesting that while they may appear to contribute useful information, they may also introduce noise or redundant signals when combined under fine-grained coordination.

In the coarse-grained setting (Table 2b), a similar but weaker pattern is observed for the Technical Agent, further supporting the importance of technical signals in the overall system. Notably, the “w/o News” row exhibits strongly negative differences across most portfolio sizes, with relatively large magnitudes. This behavior is not clearly observed in the fine-grained setting. One possible interpretation is that, in the absence of fine-grained task decomposition, news information may be relatively better utilized, potentially compensating for weaker propagation of technical signals.

Overall, the results indicate that performance depends not only on agent diversity but also on how information is structured and propagated across the system. Fine-grained task decomposition appears to facilitate more effective signal transmission—particularly for technical signals—while reducing redundancy and noise introduced by loosely coordinated agents. This highlights the importance of task design and information routing in hierarchical LLM agent architectures.

6.3 Text Analysis for Interpretability

Understanding the rationale behind LLM outputs is critical for practical deployment, especially in financial trading. Given the significant performance disparities observed between different prompting strategies in our experiments, we analyze the textual outputs generated during backtesting.

First, we compare the textual outputs generated under fine-grained and coarse-grained settings using the log-odds ratio with a Dirichlet prior [37], a statistical measure to compare how strongly a word is associated with one group versus another. Second, we analyze information propagation across agents to quantify how much information from lower-level agents is reflected in higher-level agents’ outputs. Concretely, inspired by the previous research on information propagation [e.g., 5, 75], we first convert each agent’s output text into vector representations using an LLM embedding model (text-embedding-3-small [41]). We then compute cosine similarities between agent output vectors to measure the degree of semantic alignment, which serves as a proxy for information adoption across the agent hierarchy.

6.3.1 Representative words of Fine-grained vs Coarse-grained Settings. We obtained the highest log-odds ratios for fine-grained and coarse-grained settings across four agent types: Technical, Quantitative, Sector, and PM Agents. We present the complete list of words (shown in Table 5), as well as the preprocessing of texts, in Appendix D for brevity.

First, we observe that both the fine-grained and coarse-grained configuration emphasizes vocabulary that is closely aligned with

their respective prompt instruction, which is consistent with our expectations. The fine-grained setting tends to produce more nuanced analytical terms such as “momentum,” “volatility,” and “condition” for Technical, and “margins,” “growth-rate,” and “profitability” for Quant. In contrast, the coarse-grained setting favors more surface-level market descriptors, such as “price,” “trend,” “rise,” and “increase” for Technical, and “EPS,” “earnings,” and “net income”–related expressions for Quant. This confirms that prompt granularity directly influences the level of abstraction in generated reasoning: without explicit procedural guidance, the LLM reverts to broad, superficial descriptions of market movements and financial statements.

Second, observing the hierarchical relationships, we find evidence of vocabulary propagation from lower-level specialists to higher-level decision makers. Higher-level agents, namely the Sector and PM Agents, tend to reuse or inherit vocabulary originating from lower-level agents. For instance, in the fine-grained setting, the PM and Sector Agents’ distinctive words include “Momentum” (characteristic of the Technical Agent) and “Soundness” (characteristic of the Quant Agent). Similarly, in the coarse-grained setting, the PM and Sector Agents adopt “Trend” and “EPS,” mirroring the vocabulary of their respective subordinates. This indicates that the hierarchical architecture enables upward propagation of semantic signals, suggesting that higher-level decision-making is at least partially grounded in lower-level analytical outputs.

6.3.2 Information Propagation Analysis. Table 3 shows the cosine similarity between the Sector Agent outputs and those of lower-level agents, which presents the median values of the aggregated results over 50 independent backtesting trials. The table reports similarities under both fine-grained and coarse-grained settings, along with $\text{Diff.} = \text{Similarity}_{\text{Fine-grained}} - \text{Similarity}_{\text{Coarse-grained}}$. Regarding the absolute magnitude of similarity, the Quantitative and Qualitative Agents exhibit relatively high scores ($\approx 0.48 - 0.52$) compared to the Technical Agent ($\approx 0.40 - 0.42$). This suggests that, by default, the Sector Agents’ decision-making logic aligns more closely with fundamental analysis (financials and business models) rather than technical price analysis. One might think that the length of the output of each agent affects the similarity, but we set the output length within 100 Japanese characters for each agent, and we observed no significant difference in the output lengths across agents. However, most importantly, regarding the comparison be-

the Technical Agents perform well, and their impact is critically high, especially in a fine-grained setting.

Note that, across trials, the similarity values exhibit very small variance, with below 0.002 standard deviation for all cases. Because of this small deviation, the difference between the two settings (Diff.) is statistically significant in all cases.

Taken together, while the system naturally leans towards fundamental data, the proposed fine-grained architecture effectively amplifies the signal transmission of technical analysis, indicating that technical factors are explicitly integrated into the higher-level decision-making process in the fine-grained setting.

6.4 Portfolio Optimization

In regulated financial environments, deploying a fully autonomous trading system directly to live capital is typically infeasible without extensive staged validation. Consequently, to demonstrate the real-world applicability of our system under these constraints, we perform standard portfolio optimization against a market index as a realistic pre-deployment validation setting. We conduct a systematic backtest to evaluate the allocation between the TOPIX 100 equity index and a composite portfolio of six LLM-based agent strategies. Six LLM-based agent strategies include the strategies using all agents and five leave-one-out strategies, constructed using an equal risk contribution weighting scheme. Here, we exploit the heterogeneity in outputs across the six strategies—instructed with different combinations of information sources.

The covariance structure among the six agent strategies is derived from the stock-level covariance matrix of TOPIX 100 constituents. Letting $V \in \mathbb{R}^{n \times n}$ denote the stock covariance matrix and $P \in \mathbb{R}^{M \times n}$ the portfolio weight matrix whose rows correspond to each agent’s stock-level holdings, the agent-level covariance is computed as $\Sigma = P V P^\top$. To combine the six strategies into a single composite, we solve for the weight vector w that equalizes each agent’s risk contribution to total portfolio variance: each asset’s contribution $w_i(\Sigma w)_i / \sqrt{w^\top \Sigma w}$ is driven toward $1/N$ via constrained optimization, subject to $\sum_i w_i = 1$.

We then vary the allocation ratio between TOPIX 100 and the agent composite from 0% to 100% in 10% increments, evaluating out-of-sample performance including annualized return, volatility, and Sharpe ratio, net of realistic transaction costs (10bps one-way). A key empirical finding is that the correlation between the TOPIX 100 index returns and the agent composite returns is low (≈ 0.4), creating substantial diversification benefits. As Figure 3 shows, blended portfolios consistently achieve higher Sharpe ratios than either the TOPIX 100 index or the agent composite alone (The return and volatility are shown in Table 4 in Appendix C). While the ex-ante optimal allocation ratio is unknown in practice, even a naïve 50%/50% split between the index and the agent composite yields a Sharpe ratio superior to both standalone components—demonstrating that practitioners can capture meaningful risk-adjusted performance improvements without requiring precise allocation optimization.

7 Discussion and Conclusion

In this study, we constructed a hierarchical multi-agent trading framework and investigated how task granularity affects system

Table 3: Semantic Similarity with Sector Agent

Agents	Fine-grained	Coarse-grained	Diff.
Technical	0.419	0.397	0.022
Quantitative	0.476	0.477	-0.001
Qualitative	0.514	0.514	-0.001
News	0.378	0.372	0.006

tween the fine-grained and coarse-grained settings, we find that only the Technical Agent demonstrates a significant improvement in the fine-grained setting (0.022 in Diff.). Specifically, in the fine-grained setting, the higher similarity score suggests that technical insights are effectively transmitted and integrated into the Sector Agent’s reasoning process. This aligns with the backtesting results;

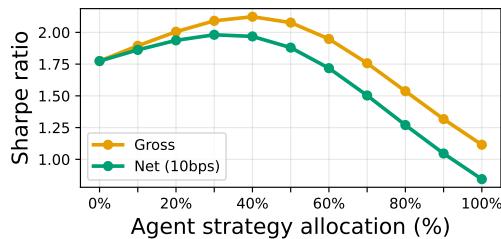


Figure 3: Sharpe ratio as a function of allocation between TOPIX 100 and the aggregated agent strategy in the test period. Gross performance (orange) and net performance after 10 bps one-way transaction cost (green) are shown.

behavior by comparing fine-grained and coarse-grained task settings. Our experimental results demonstrate that the fine-grained setting yields statistically superior overall performance in terms of risk-adjusted returns. Furthermore, the ablation study revealed that the *Technical* agent is a primary performance driver. Crucially, text-based analyses consistently confirmed that the fine-grained instructions enabled the effective propagation of technical insights to higher-level decision-makers. These results provide evidence from performance, ablation, and textual behavior analyses that fine-grained task structuring improves both the effectiveness and the information flow within hierarchical LLM agent systems. Finally, we evaluated real-world performance through portfolio optimization using market indices as benchmarks. Our findings suggest that the performance of LLM-based trading agents is driven not merely by the model’s reasoning capability, but significantly by the quality of feature engineering embedded within the prompt design.

7.1 Implications

From an agent design perspective, our findings suggest a shift in how to construct multi-agent systems for financial analysis. Prior work has often implicitly assumed a one-to-one mapping between data modality and agent specialization, with the role-based ambiguous instructions. However, our results indicate that agents may be more effectively designed around task decomposition rather than data source boundaries. In practical settings, this opens the possibility that users can embed their own domain-specific expertise directly into task-specialized agents, enabling customizable and organization-specific agent frameworks.

Another important implication concerns interpretability. Our study demonstrates that meaningful insights can be extracted from analyzing agent text outputs, providing a practical pathway for understanding LLM-driven decision processes. Interpretability is especially critical in enterprise settings, particularly for large-scale asset management. Prior work has suggested that the adoption of LLMs introduces new operational workflows centered around validating generated outputs [19]. In this context, building trading agents with strong interpretability characteristics is not merely desirable but may become operationally necessary.

Simultaneously, there is an ongoing debate over whether natural language should be adopted as the primary communication interface in multi-agent LLM systems. Many existing frameworks [27, 45,

69, 76] rely on natural language communication, while alternative approaches propose machine-oriented languages that are mutually intelligible among AI agents to improve efficiency and accuracy [63, 73]. Nevertheless, from a practical perspective [18], natural language interfaces appear advantageous, as they enable interpretability and downstream analyses such as those conducted in our study.

7.2 Limitations and Future Work

Despite promising results, several limitations remain. First, it is not yet fully clear whether the performance gains are fundamentally attributable to fine-grained task decomposition itself. One alternative explanation is that certain vocabulary patterns may be more easily adopted by the preference of LLMs to influence downstream agents. Investigating linguistic bias in LLM-based multi-agent systems is, therefore, an important direction for future research [c.f., 17, 26].

Second, due to the knowledge cutoff of the LLM model, back-testing was limited to approximately two years of historical data. Financial markets exhibit strong regime shifts over longer horizons, and therefore, longer-term validation is necessary to confirm robustness. One possible future direction is the use of time-aware or temporally constrained LLM variants, such as approaches similar to Time Machine GPT [10], which could enable historically consistent simulations across longer market periods.

Third, while we established rigorous experimental settings, we acknowledge that there remains room for exploration under different conditions, such as employing other LLM models or targeting different markets (e.g., the US market). However, we believe the scope of this analysis is sufficient for the following reasons. Primarily, from an industrial perspective, our primary focus is on the Japanese market due to specific deployment requirements. Also, current LLMs have demonstrated sufficient pre-trained knowledge regarding the Japanese market [40]. Furthermore, given the recent competitive progress across providers [54], we infer that the fundamental validity of our proposed method remains consistent across high-performing models.

References

- [1] Saaket Agashe, Jiuzhou Han, Shuyu Gan, Jiachen Yang, Ang Li, and Xin Eric Wang. 2024. Agent s: An open agentic framework that uses computers like a human. *arXiv preprint arXiv:2410.08164* (2024).
- [2] Ran Aroussi. [n. d.]. ranaroussi/yfinance: Download market data from Yahoo! Finance’s API. <https://github.com/ranaroussi/yfinance?tab=readme-ov-file> [Online; accessed 2026-02-05].
- [3] Berk Atil, Sarp Aykent, Alexa Chittams, Lisheng Fu, Rebecca J Passonneau, Evan Radcliffe, Guru Rajan Rajagopal, Adam Sloan, Tomasz Tudej, Ferhan Ture, et al. 2024. Non-determinism of “deterministic” LLM settings. *arXiv preprint arXiv:2408.04667* (2024).
- [4] Erik Brynjolfsson, Danielle Li, and Lindsey Raymond. 2025. Generative AI at work. *The Quarterly Journal of Economics* 140, 2 (2025), 889–942.
- [5] Tristan JB Cann, Ben Dennes, Travis Coan, Saffron O’Neill, and Hywel TP Williams. 2025. Using semantic similarity to measure the echo of strategic communications. *EPJ Data Science* 14, 1 (2025), 20.
- [6] Ceek.jp. [n. d.]. Ceek.jp News. <https://news.ceek.jp/> [Online; accessed 2026-02-05].
- [7] Jiaxiang Chen, Mingxi Zou, Zhuo Wang, Qifan Wang, Dongning Sun, Chi Zhang, and Zenglin Xu. 2025. FinHEAR: Human Expertise and Adaptive Risk-Aware Temporal Reasoning for Financial Decision-Making. *arXiv preprint arXiv:2506.09080* (2025).
- [8] Mark Chen. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374* (2021).
- [9] Yifei Dong, Fengyi Wu, Kunlin Zhang, Yilong Dai, Sanjian Zhang, Wanghao Ye, Sihan Chen, and Zhi-Qi Cheng. 2025. Large Language Model Agents in Finance: A Survey Bridging Research, Practice, and Real-World Deployment. In *Findings of the Association for Computational Linguistics: EMNLP 2025*. 17889–17907.

- [10] Felix Drinkall, Eghbal Rahimikia, Janet Pierrehumbert, and Stefan Zohren. 2024. Time machine GPT. In *Findings of the Association for Computational Linguistics: NAACL 2024*. 3281–3292.
- [11] EDINET. [n. d.]. EDINET. <https://disclosure2d.edinet-fsa.go.jp/guide/static/disclosure/WEEK0060.html> [Online; accessed 2026-02-05].
- [12] FRED. [n. d.]. St. Louis Fed Web Services: FRED® API. <https://fred.stlouisfed.org/docs/api/fred/> [Online; accessed 2026-02-05].
- [13] Zhixuan He and Yue Feng. 2025. Unleashing Diverse Thinking Modes in LLMs through Multi-Agent Collaboration. *arXiv preprint arXiv:2510.16645* (2025).
- [14] Sirui Hong, Mingchen Zhuge, Jonathan Chen, Xiaowu Zheng, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, Zili Wang, Steven Ka Shing Yau, Zijuan Lin, et al. 2023. MetaGPT: Meta programming for a multi-agent collaborative framework. In *The Twelfth International Conference on Learning Representations*.
- [15] Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. 2024. GPT-40 system card. *arXiv preprint arXiv:2410.21276* (2024).
- [16] Yoontae Hwang, Yaxuan Kong, Stefan Zohren, and Yongjae Lee. 2025. Decision-informed neural networks with large language model integration for portfolio optimization. *arXiv preprint arXiv:2502.00828* (2025).
- [17] Yerin Hwang, Dongryeo Lee, Taegwan Kang, Minwoo Lee, and Kyomin Jung. 2026. When Wording Steers the Evaluation: Framing Bias in LLM judges. *arXiv preprint arXiv:2601.13537* (2026).
- [18] Aakanksha Jadhav and Vishal Mirza. 2025. Large Language Models in Equity Markets: Applications, Techniques, and Insights. *Frontiers in Artificial Intelligence* Volume 8 - 2025 (2025).
- [19] Ranim Khojah, Mazen Mohamad, Linda Erlenov, Francisco Gomes de Oliveira Neto, and Philipp Leitner. 2025. LLM Company Policies and Policy Implications in Software Organizations. *IEEE Software* (2025).
- [20] Alex Kim, Maximilian Muhn, and Valeri Nikolaeve. 2024. Financial statement analysis with large language models. *arXiv preprint arXiv:2407.17866* (2024).
- [21] Hyuhng Joon Kim, Youna Kim, Cheonbok Park, Junyeob Kim, Choonghyun Park, Kang Min Yoo, Sang-goo Lee, and Taetuk Kim. 2024. Aligning language models to explicitly handle ambiguity. *arXiv preprint arXiv:2404.11972* (2024).
- [22] Harvey Bonmu Ku, Jeongyeol Shin, Hyoun Jun Lee, Seonok Na, and Insu Jeon. 2025. Multi-agent LLM debate unveils the premise left unsaid. In *Proceedings of the 12th Argument mining Workshop*. 58–73.
- [23] Taku Kudo. 2005. Mecab: Yet another part-of-speech and morphological analyzer. <http://mecab.sourceforge.net/> (2005).
- [24] Sayani Kundu, Dushyant Sahoo, Victor Li, Jennifer Rabowsky, and Amit Varshney. 2025. A Multi-Agent Framework for Quantitative Finance: An Application to Portfolio Management Analytics. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: Industry Track*. 812–824.
- [25] Charles MC Lee. 2025. Value investing: integrating theory and practice. In *Handbook on the Financial Reporting Environment*. Edward Elgar Publishing, 347–374.
- [26] Hoyoung Lee, Junhyuk Seo, Suhwan Park, Junhyeong Lee, Wonbin Ahn, Chanyeol Choi, Alejandro Lopez-Lira, and Yongjae Lee. 2025. Your AI, not your view: The bias of llms in investment analysis. In *Proceedings of the 6th ACM International Conference on AI in Finance*. 150–158.
- [27] Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbulin, and Bernard Ghanem. 2023. Camel: Communicative agents for “mind” exploration of large language model society. *Advances in Neural Information Processing Systems* 36 (2023), 51991–52008.
- [28] Miao Li, Jey Han Lau, Eduard Hovy, and Mirella Lapata. 2025. Decomposed Opinion Summarization with Verified Aspect-Aware Modules. *arXiv preprint arXiv:2501.17191* (2025).
- [29] Xiangyu Li, Yawen Zeng, Xiaofen Xing, Jin Xu, and Xiangmin Xu. 2025. Hedgeagents: A balanced-aware multi-agent financial trading system. In *Companion Proceedings of the ACM on Web Conference 2025*. 296–305.
- [30] Yawei Li, Peipei Liu, and Ze Wang. 2022. Stock trading strategies based on deep reinforcement learning. *Scientific Programming* 2022, 1 (2022), 4698656.
- [31] Michael Lin. 2019. Quantitative vs. fundamental equity investing: Active Quantitative Equity (AQE).
- [32] Xiangdong Liu and Jiahao Chen. 2025. QTMLR: An Agent for Quantitative Trading Decision-Making Based on Multi-Indicator Guided Reinforcement Learning. *arXiv preprint arXiv:2508.20467* (2025).
- [33] Joshua Livnat and Richard R Mendenhall. 2006. Comparing the post-earnings announcement drift for surprises calculated from analyst and time series forecasts. *Journal of accounting research* 44, 1 (2006), 177–205.
- [34] Alejandro Lopez-Lira. 2025. Can Large Language Models Trade? Testing Financial Theories with LLM Agents in Market Simulations. *arXiv preprint arXiv:2504.10789* (2025).
- [35] Alejandro Lopez-Lira, Yuehua Tang, and Mingyin Zhu. 2025. The Memorization Problem: Can We Trust LLMs’ Economic Forecasts? *arXiv preprint arXiv:2504.14765* (2025).
- [36] Aaron Mok. 2023. Wharton Professor Says AI Is Like an ‘Intern’ Who ‘Lies a Little Bit’ - Business Insider. <https://www.businessinsider.com/wharton-professor-ai-is-intern-who-lies-a-little-bit-2023-5> [Online; accessed 2025-11-28].
- [37] Burt L Monroe, Michael P Colaresi, and Kevin M Quinn. 2008. Fightin’words: Lexical feature selection and evaluation for identifying the content of political conflict. *Political Analysis* 16, 4 (2008), 372–403.
- [38] Sina Montazeri, Haseebullah Jumakhan, and Amir Mirzaeinia. 2025. Finding Optimal Trading History in Reinforcement Learning for Stock Market Trading. *arXiv preprint arXiv:2502.12537* (2025).
- [39] Yuqi Nie, Yaxuan Kong, Xiaowen Dong, John M Mulvey, H Vincent Poor, Qingsong Wen, and Stefan Zohren. 2024. A survey of large language models for financial applications: Progress, prospects and challenges. *arXiv preprint arXiv:2406.11903* (2024).
- [40] Katsuhiko Okada, Moe Nakasuiji, Yasutomo Tsukioka, and Takahiro Yamasaki. 2025. From words to returns: sentiment analysis of Japanese 10-K reports using advanced large language models. *PeerJ Computer Science* 11 (2025), e3349.
- [41] OpenAI. [n. d.]. text-embedding-3-small Model | OpenAI API. <https://platform.openai.com/docs/models/text-embedding-3-small> [Online; accessed 2026-02-08].
- [42] Shuyin Ouyang, Jie M Zhang, Mark Harman, and Meng Wang. 2025. An empirical study of the non-determinism of chatgpt in code generation. *ACM Transactions on Software Engineering and Methodology* 34, 2 (2025), 1–28.
- [43] Avash Palikhe, Zhenyu Yu, Zichong Wang, and Wenbin Zhang. 2025. Towards Transparent AI: A Survey on Explainable Large Language Models. *arXiv preprint arXiv:2506.21812* (2025).
- [44] George Papageorgiou, Dimitrios Gkaimanis, and Christos Tjortjis. 2024. Enhancing stock market forecasts with double deep q-network in volatile stock market environments. *Electronics* 13, 9 (2024), 1629.
- [45] Joon Sung Park, Joseph O’Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*. 1–22.
- [46] Sida Peng, Eirini Kalliamvakou, Peter Cihon, and Mert Demirer. 2023. The impact of AI on developer productivity: Evidence from github copilot. *arXiv preprint arXiv:2302.06590* (2023).
- [47] Stephen H Penman. 2010. *Financial statement analysis and security valuation*. McGraw-Hill/Irwin New York.
- [48] Joseph D Piotroski. 2000. Value investing: The use of historical financial statement information to separate winners from losers. *Journal of accounting research* (2000), 1–41.
- [49] Libin Qiu, Yuhang Ye, Zhirong Gao, Xide Zou, Junfu Chen, Ziming Gui, Weizhi Huang, Xiaobo Xue, Wenkai Qiu, and Kun Zhao. 2025. Blueprint First, Model Second: A Framework for Deterministic LLM Workflow. *arXiv preprint arXiv:2508.02721* (2025).
- [50] Evan Ratliff. 2025. All of My Employees Are AI Agents, and So Are My Executives | WIRED. <https://www.wired.com/story/all-my-employees-are-ai-agents-so-are-my-executives/> [Online; accessed 2025-11-28].
- [51] Preetha Saha, Jingrui Lyu, Arnav Saxena, Tianjiao Zhao, and Dhagash Mehta. 2025. Large Language Model Agents for Investment Management: Foundations, Benchmarks, and Research Frontiers. In *Proceedings of the 6th ACM International Conference on AI in Finance*. 736–744.
- [52] Parshin Shojaee, Iman Mirzadeh, Keivan Alizadeh, Maxwell Horton, Samy Bengio, and Mehrdad Farajtabar. 2025. The illusion of thinking: Understanding the strengths and limitations of reasoning models via the lens of problem complexity. *arXiv preprint arXiv:2506.06941* (2025).
- [53] Alex Singla, Alexander Sukharevsky, Lareina A. Yee, and Michael Chui. 2025. The State of AI: Global Survey 2025 | McKinsey. <https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-state-of-ai> [Online; accessed 2025-11-28].
- [54] AI Wiki Editorial Team. 2025. LLM Benchmark Rankings 2025 | AI Model Comparison Guide. <https://artificial-intelligence-wiki.com/generative-ai/large-language-models/llm-benchmark-rankings-2025> [Online; accessed 2026-02-07].
- [55] Feng Tian, Flora D Salim, and Hao Xue. 2025. TradingGroup: A Multi-Agent Trading System with Self-Reflection and Data-Synthesis. *arXiv preprint arXiv:2508.17565* (2025).
- [56] Taiichi Hashimoto Toshinori Sato and Manabu Okumura. 2017. Implementation of a word segmentation dictionary called mecab-ipadic-NEologd and study on how to use it effectively for information retrieval (in Japanese). In *Proceedings of the Twenty-three Annual Meeting of the Association for Natural Language Processing*. The Association for Natural Language Processing, NLP2017–B6–1.
- [57] Lei Wang, Wanyu Xu, Yihuai Lan, Zhiqiang Hu, Yunshi Lan, Roy Ka-Wei Lee, and Ee-Peng Lim. 2023. Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. *arXiv preprint arXiv:2305.04091* (2023).
- [58] Saizhuo Wang, Hang Yuan, Leon Zhou, Lionel Ni, Heung Yeung Shum, and Jian Guo. 2025. Alpha-gpt: Human-ai interactive alpha mining for quantitative investment. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. 196–206.
- [59] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171* (2022).

- , ,
- [60] Jiaxin Wen, Ruiqi Zhong, Akbir Khan, Ethan Perez, Jacob Steinhardt, Minlie Huang, Samuel R Bowman, He He, and Shi Feng. 2024. Language models learn to mislead humans via RLHF. *arXiv preprint arXiv:2409.12822* (2024).
- [61] Zeyu Xia, Mingde Shi, and Changle Lin. 2023. Stock trading strategy developing based on reinforcement learning. In *Proceedings of the 2nd International Academic Conference on Blockchain, Information Technology and Smart Finance (ICBIS 2023)*. Atlantis Press, 156–164.
- [62] Yijia Xiao, Edward Sun, Di Luo, and Wei Wang. 2024. TradingAgents: Multi-agents LLM financial trading framework. *arXiv preprint arXiv:2412.20138* (2024).
- [63] Zhuoran Xiao, Chenhui Ye, Yijia Feng, Yunbo Hu, Tianyu Jiao, Liyu Cai, and Guangyi Liu. 2025. Transmission With Machine Language Tokens: A Paradigm for Task-Oriented Agent Communication. *arXiv preprint arXiv:2507.21454* (2025).
- [64] Fei Xiong, Xiang Zhang, Aosong Feng, Siqi Sun, and Chenyu You. 2025. Quantagent: Price-driven multi-agent llms for high-frequency trading. *arXiv preprint arXiv:2509.09995* (2025).
- [65] Hongyang Yang, Boyu Zhang, Neng Wang, Cheng Guo, Xiaoli Zhang, Likun Lin, Junlin Wang, Tianyu Zhou, Mao Guan, Runjia Zhang, et al. 2024. FinRobot: An open-source ai agent platform for financial applications using large language models. *arXiv preprint arXiv:2405.14767* (2024).
- [66] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. In *The eleventh international conference on learning representations*.
- [67] Yangyang Yu, Haohang Li, Zhi Chen, Yuechen Jiang, Yang Li, Jordan W Suchow, Denghui Zhang, and Khalidou Khashanah. 2025. Finmem: A performance-enhanced LLM trading agent with layered memory and character design. *IEEE Transactions on Big Data* (2025).
- [68] Yangyang Yu, Zhiyuan Yao, Haohang Li, Zhiyang Deng, Yuechen Jiang, Yupeng Cao, Zhi Chen, Jordan Suchow, Zhenyu Cui, Rong Liu, et al. 2024. Fincon: A synthesized lln multi-agent system with conceptual verbal reinforcement for enhanced financial decision making. *Advances in Neural Information Processing Systems 37* (2024), 137010–137045.
- [69] Ceyao Zhang, Kajie Yang, Siyi Hu, Zihao Wang, Guanghe Li, Yihang Sun, Cheng Zhang, Zhaowei Zhang, Anji Liu, Song-Chun Zhu, et al. 2024. Proagent: building proactive cooperative agents with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 17591–17599.
- [70] Wentao Zhang, Lingxuan Zhao, Haochong Xia, Shuo Sun, Jiaze Sun, Molei Qin, Xinyi Li, Yuqing Zhao, Yilei Zhao, Xinyu Cai, et al. 2024. A multimodal foundation agent for financial trading: Tool-augmented, diversified, and generalist. In *Proceedings of the 30th ACM SIGKDD conference on knowledge discovery and data mining*. 4314–4325.
- [71] Haiyan Zhao, Hanjie Chen, Fan Yang, Ninghao Liu, Huiqi Deng, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, and Mengnan Du. 2024. Explainability for large language models: A survey. *ACM Transactions on Intelligent Systems and Technology* 15, 2 (2024), 1–38.
- [72] Tianjiao Zhao, Jingrao Lyu, Stokes Jones, Harrison Garber, Stefano Pasquali, and Dhagash Mehta. 2025. AlphaAgents: Large Language Model based Multi-Agents for Equity Portfolio Constructions. *arXiv preprint arXiv:2508.11152* (2025).
- [73] Yujia Zheng, Zhuokai Zhao, Zijian Li, Yaqi Xie, Mingze Gao, Lizhu Zhang, and Kun Zhang. 2025. Thought communication in multiagent collaboration. *arXiv preprint arXiv:2510.20733* (2025).
- [74] Denny Zhou, Nathanael Schärlí, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, et al. 2022. Least-to-most prompting enables complex reasoning in large language models. *arXiv preprint arXiv:2205.10625* (2022).
- [75] Jialong Zhou, Lichao Wang, and Xiao Yang. 2025. GUARDIAN: Safeguarding LLM Multi-Agent Collaborations with Temporal Graph Modeling. *arXiv preprint arXiv:2505.19234* (2025).
- [76] Mingchen Zhuge, Haozhe Liu, Francesco Faccio, Dylan R Ashley, Róbert Csordás, Anand Gopalakrishnan, Abdullah Hamdi, Hasan Abed Al Kader Hammoud, Vincent Herrmann, Kazuki Irie, et al. 2023. Mindstorms in natural language-based societies of mind. *arXiv preprint arXiv:2305.17066* (2023).

A Formulas for Technical Indicators

A.1 MACD

We calculate the MACD line ($M_t = \text{EMA}_{12} - \text{EMA}_{26}$), the signal line ($S_t = \text{EMA}_9(M_t)$), and the histogram ($H_t = M_t - S_t$), where EMA indicates the exponential moving average with the smoothing factor $\alpha = \frac{2}{t+1}$. These values are normalized by the closing price P_t (i.e., M_t/P_t) to enable cross-sectional comparison.

A.2 RSI

The RSI is defined as

$$RSI_t = 100 - \frac{100}{1 + RS_t},$$

where

$$RS_t = \frac{\text{AvgGain}_t}{\text{AvgLoss}_t}.$$

The average gain and loss are computed using exponentially smoothed moving averages over a 14-day lookback period.

A.3 Stochastic Oscillator Formulation

Let P_t denote the closing price at time t . We define the highest and lowest prices over the past n days as

$$H_t^{(n)} = \max_{i=0, \dots, n-1} P_{t-i}, \quad (1)$$

$$L_t^{(n)} = \min_{i=0, \dots, n-1} P_{t-i}. \quad (2)$$

The stochastic oscillator $\%K$ is defined as

$$\%K_t = 100 \times \frac{P_t - L_t^{(9)}}{H_t^{(9)} - L_t^{(9)}}. \quad (3)$$

The signal line $\%D$ is computed as the 3-day simple moving average (SMA) of $\%K$:

$$\%D_t = \frac{1}{3} \sum_{i=0}^2 \%K_{t-i}. \quad (4)$$

Finally, the divergence term J is defined as

$$J_t = 3\%D_t - 2\%K_t. \quad (5)$$

B Prompts

B.1 Technical Agent

B.1.1 System Prompt.

System Prompt (Role Definition):

Role: You are a technical analyst on the trading team. Your task is to forecast stock prices one month ahead based strictly on technical indicators to assist portfolio managers.

Policy & Constraints:

- **Input scope:** Use only the provided technical indicators; disregard news or fundamentals.
- **Scoring:** Provide a score between 0 and 100 based on a balanced assessment of momentum, oscillators, and volatility.
- **Scale interpretation:**
 - 100: Strong long recommendation
 - 50: Neutral (no clear advantage)
 - 0: Strong short recommendation
- **Output requirements:** strictly JSON format, including a brief one-sentence comment.

B.1.2 User Prompt (Fine-Grained).

Instruction: The following are technical indicators for a particular stock at the end of a given month. Based on these, please rate the attractiveness of long or short positions in this stock on a scale of 0 to 100 points.

Technical indicators used (summary of definitions/reference):

- **Momentum: RoC (Rate of Change)**
 - Percentage change in price over the past n days ($n = 5, 10, 20$, compared to previous month) and m months ($n = 1, 3, 6, 12$).
- **Volatility: Bollinger Band Deviation**
 - Actual value used: $(\text{Close} - 20\text{-day MA}) / 20\text{-day Close standard deviation}$
- **Oscillators:**

- **MACD:** (12-day EMA - 26-day EMA) / Close price. Normalized by dividing by Close.
- **RSI:** Ranges from 0 to 100. $(100 \times \text{Upward Avg}) / (\text{Upward Avg} + \text{Downward Avg})$.
- **Stochastic Oscillator:** Uses K, D, J indicators.

Evaluation Rules:

- Comprehensively assess the risk-reward ratio for the next month based on the combination of indicators.
- Check consistency among momentum, oscillators, and volatility.
- **100:** Strong Long, **0:** Strong Short, **50:** Neutral.
- State the reason in Japanese in one sentence (≈ 50 chars).

Output Format (JSON only):

```
{
  "score": <integer 0-100>,
  "reason": "<explanation>"
}
```

This Month's Technical Indicators:

```

RoC 5day: <value>%   RoC 10day: <value>%
RoC 20day: <value>%   RoC 1Month: <value>%
RoC 3Month: <value>%   RoC 6Month: <value>%
RoC 12Month: <value>%
RSI: <value>
MACD: <value>   Signal: <value>   Hist: <value>
Stochastic %K: <value>   %D: <value>   %J: <value>
```

B.1.3 User Prompt (Coarse-Grained).

Instruction: The following list contains raw daily closing prices for a particular stock over the past 252 business days. The values on the left are the most recent. Based on these, rate the attractiveness of long or short positions on a scale of 0 to 100.

Evaluation Rules:

- Analyze the price trend to assess the risk-reward ratio for the next month.
- **100:** Strong Long, **50:** Neutral, **0:** Strong Short.
- State the reason in Japanese (≈ 50 chars).

Output Format (JSON only):

```
{
  "score": <integer 0-100>,
  "reason": "<explanation>"
}
```

This Month's Stock Prices (List of 252 days):

```
[<price_t>, <price_t-1>, ..., <price_t-251>]
(e.g., [1500.5, 1498.2, ..., 1200.0])
```

B.2 Quant. Agent

B.2.1 System Prompt.

System Prompt (Role Definition):

Role: You are a Quantitative Fundamental Analyst. Your task is to evaluate the medium-to-long-term investment attractiveness of a stock based strictly on quantitative financial metrics to assist the Portfolio Manager.

Guidelines & Constraints:

- **Input scope:** Use only the provided financial metrics; exclude news, sentiment, or technical patterns.
- **Evaluation Balance:** Assess Profitability (Margins, ROE), Value (PER), Financial Health (Quick Ratio, D/E), Growth, and Cash Flow quality.
- **Scoring Scale:**
 - **100:** Strong Long (extremely attractive)
 - **50:** Neutral (fairly valued)
 - **0:** Strong Short (extremely unattractive)
- **Missing Data:** Ignore items marked "NaN" or blank; analyze based on remaining data.
- **Output requirements:** Strictly JSON format; "reason" must be a single short sentence in Japanese.

B.2.2 User Prompt (Fine-Grained).

Instruction: The following are fundamental metrics and their changes from a month ago. Evaluate the attractiveness for a Long/Short position on a scale of 0 to 100 based on these.

Rules & Definitions:

- **Metrics:** Profitability, Value (PER), Cash Flow, Financial Health, Growth (Sales/EPS).
- **Trend Analysis:** Consider both "Absolute Value" and "Diff" (change from prev. month).
- **Freshness:** If "Information Update Month" is "Yes", latest results are reflected.
- **Output:** Score (0-100) and reason (in Japanese, ≈ 50 chars).

Output Format (JSON only):

```
{
  "score": <integer 0-100>,
  "reason": "<explanation>"
}
```

Stock Data (TTM) [Format: Value (diff: Value)]:

```

Info Update Month: <Yes/No>
[Profitability] Net Margin: <val> (diff: <val>)   ROA: <val>
(dif: <val>)
ROE: <val> (diff: <val>)   Asset Turn: <val> (diff: <val>)
Inv. Turn Days: <val> (diff: <val>)
[Value] PER: <val> (diff: <val>)
[Cash Flow] FCF: <val> (diff: <val>)   Margin: <val> (diff: <val>)
EBITDA: <val> (diff: <val>)
[Health] Equity Ratio: <val> (diff: <val>)   Quick Ratio: <val>
(dif: <val>)
D/E Ratio: <val> (diff: <val>)
[Growth] Sales YoY: <val> (diff: <val>)   CAGR 3Y: <val> (diff: <val>)
EPS Growth: <val> (diff: <val>)   DPS: <val> (diff: <val>)
```

B.2.3 User Prompt (Coarse-Grained).

Instruction: Evaluate the attractiveness of this stock for a Long/Short position (0-100) based on the fundamental metrics and their changes (RoC) from a month ago.

Rules & Format:

- **Trend Analysis:** Consider both "Absolute Value" and "RoC" (Rate of Change).
- **Handling Missing Data:** Judge based on available info.
- **Output:** JSON format with a score (0-100) and a Japanese reason (≈ 50 chars).

Stock Data (TTM) [Format: Value (RoC: Value%)]:

```

Info Update: <Yes/No>
[P/L] Sales: <val> (RoC: <val>)   Cost of Sales: <val> (RoC: <val>)
Op Profit: <val> (RoC: <val>)   Net Income: <val> (RoC: <val>)
Depreciation: <val> (RoC: <val>)
[EPS] Current: <val> (RoC: <val>)   1y Ago: <val>   3y Ago: <val>
[B/S: Assets] Total Assets: <val> (RoC: <val>)   Cash: <val> (RoC: <val>)
Receivables: <val> (RoC: <val>)   Inventory: <val> (RoC: <val>)
Financial Assets: <val> (RoC: <val>)
[B/S: Liab/Eq] Equity: <val> (RoC: <val>)   Debt: <val> (RoC: <val>)
Cur. Liabilities: <val> (RoC: <val>)
[Cash Flow] Op CF: <val> (RoC: <val>)   Inv CF: <val> (RoC: <val>)
[Others] Dividends: <val> (RoC: <val>)   Issued Shares: <val> (RoC: <val>)
Monthly Close: <val> (RoC: <val>)
```

B.3 Qual. Agent

B.3.1 System Prompt.

System Prompt (Role Definition):

Role: You are a Strategic Analyst reporting to the Portfolio Manager. Your mission is to analyze qualitative corporate disclosures and provide a "Fundamental Risk & Catalyst Report" for the upcoming 1-month horizon.

Perspective & Analysis Logic:

- **Filter:** Distinguish between "stagnant boilerplate text" and "meaningful strategic shifts."
- **Focus:** Identify qualitative triggers (catalysts or red flags) rather than just long-term value.
- **Target:** Operational momentum, management credibility, and hidden structural risks.

Guidelines:

- **Inputs:** Excerpts from Securities Reports (Business Overview, Risks, MD&A, Governance).
- **Outputs:** Three specific scores (1-5) and a strategic summary ("Insight").
- **Format:** Return ONLY a JSON object. The "insight" must be written in Japanese.

B.3.2 User Prompt.

Instruction: Evaluate qualitative corporate data to advise the PM on stock attractiveness and potential risks for the next 1 month.

Evaluation Items (Score 1-5):

1. **Business Momentum:** Strength of cycle/strategy.
(1: Deteriorating/Vague → 5: Strong tailwinds/Clear execution)
2. **Immediate Risk Severity:** Probability of risks manifesting.
(1: High risk/Urgent → 5: Low risk/Stable)
3. **Management Trust:** Credibility & oversight structure.
(1: Untrustworthy → 5: Transparent/Aligned)

Rules & Output:

- **Focus:** Look for "Changes" in tone or new risk factors.
- **Insight:** Professional briefing in Japanese (~ 150 chars).
- **Format:** JSON with scores and insight.

Input Data (Text Excerpts):

- **Info Update:** <Yes/No>
- [1. Overview] <Business Description text...>
- [2. Risks] <Business Risks text...>
- [3. MD&A] <Financial Analysis text...>
- [4. Governance] <Officers/Board text...>

B.4 News Agent**B.4.1 System Prompt.****System Prompt (Role Definition):**

Role: You are a Senior News Analyst specializing in the stock market. Your task is to analyze news headlines and summaries from the past month to provide qualitative insights that complement fundamental scores.

Evaluation Guidelines:

- **Perspectives:** Evaluate impact on "Return Outlook" (Upside) and "Risk Outlook" (Downside).
- **Scoring Scale (1-5):**
· 1: Minimal/None → 3: Moderate → 5: Extreme
- **Analysis Logic:** Distinguish between temporary noise and structural changes (e.g., product launches, regulations, ESG).
- **Output:** JSON format only. Reason must be a concise Japanese summary.

B.4.2 User Prompt.

Instruction: Evaluate "Return Outlook" and "Risk Outlook" (1-3 months) based on the provided news articles.

Evaluation Criteria (Score 1-5):

- **Return Outlook:** Positive momentum (e.g., new products, expansion).
- **Risk Outlook:** Potential downside/uncertainty (e.g., supply chain, lawsuits).

Rules & Output:

- **Balance:** Identify risks even if news is generally positive.
- **Empty Case:** If no news, set both scores to 1 and reason "No News".
- **Format:** JSON with scores (1-5) and Japanese reason (~ 100 chars).

News List for the Month (Input Data):

- <List of [Date] Headline / Summary...>
- (e.g., 2024-01-15: Launched new EV model...)
- (e.g., 2024-01-20: CEO announced resignation...)

B.5 Sector Agent**B.5.1 System Prompt.****System Prompt (Role Definition):**

Role: You are a Sector Specialist on the investment committee. Your task is to synthesize reports from Technical, Quantitative, and Qualitative sub-analysts to provide a definitive 1-month investment recommendation.

Synthesis Logic & Perspective:

- **The "Bridge":** Connect raw multi-angle analysis to PM execution.
- **Metaphor:** Tech/Quant act as the "Engine" (price/value); Qualitative acts as the "Steering" (hazards).
- **Dynamic Weighting:** Adjust weights based on consistency and sector environment (e.g., high volatility → prioritize risk).

Guidelines:

- **Sector Context:** Compare metrics against sector averages to identify Leaders vs. Laggards.
- **Output:** Final Conviction Score (0-100) and a comprehensive Investment Thesis.

B.5.2 User Prompt. We alter the prompts depending on the granularity of settings.

Instruction: As the Sector Specialist, review the analyst reports and sector data to provide a final recommendation (Conviction Score & Investment Thesis).

1. Sub-Analyst Reports (Inputs):

- **Technical Analyst:** <Score & Comment>
- **Quant Fundamental Analyst:** <Score & Comment>
- **Qualitative Strategic Analyst:** <Score & Comment>

2. Sector & Comparative Context (Variable Inputs): The inputs below switch between Coarse-grained and Fine-grained settings.**[Setting A: Coarse-grained (Financial Ratios)]**

Compare Target vs. Sector Avg using high-level metrics:

- **Profitability:** Net Margin, ROA, ROE, Asset Turnover, Inv. Turn Days.
- **Value:** PER.
- **Cash Flow:** FCF, FCF Margin, EBITDA.
- **Health:** Equity Ratio, Quick Ratio, D/E Ratio.
- **Growth:** Sales YoY/CAGR, EPS YoY/3y-Ago, DPS.

[Setting B: Fine-grained (Raw RoC)]

Compare Target vs. Sector Avg using Rate of Change (RoC) for specific items:

- **P/L Items:** Sales, Op Profit, Net Income, Cost of Sales, Depreciation.
- **B/S Items:** Total Assets, Equity, Cash, Receivables, Inventory, Financial Assets, Interest Bearing Debt, Cur. Liabilities, Issued Shares.
- **CF & Others:** Op CF, Investing CF, Dividends, Monthly Close.

3. Tasks for PM Report (Output):

1. **Conviction Score (0-100):** Integrate views & sector strength. (100: Outperform, 0: Underperform).

2. **Comprehensive Thesis:** Synthesize alignment/conflict between Tech/Fund/Sector. Highlight catalysts/risks. (~ 200 words in Japanese).

Output Format: JSON with "score" and "investment_thesis".

B.6 Macro Agent**B.6.1 System Prompt.****System Prompt (Role Definition):**

Role: You are a Macro Analyst on the trading team. Your task is to analyze JP/US macro indicators to identify factors influencing the 1-month return of Japanese equities.

Evaluation Areas (Label & Score 0-100):

- **Market Direction:** Bullish/Bearish (Overall outlook).
- **Risk:** Risk sentiment and potential volatility.
- **Economy:** Economic growth trends.
- **Rates:** Interest rate environment.
- **Inflation:** Price level trends.

Policy & Constraints:

- **Input Scope:** Use only provided indicators; do not interpret news.
- **Scoring Logic:** Based on "Levels" and "Rate of Change" of indicators.
- **Output Requirement:** Strictly JSON format.
- **Summary:** Concise comment in Japanese (~ 200 chars).

B.6.2 User Prompt.

Instruction: Evaluate the current macroeconomic environment to impact the 1-month forward return of Japanese stocks, based on "Levels" and "RoC" (Rate of Change).

Evaluation Items & Scoring Rules (0-100): Score 100: Strong Buy (Bullish), 50: Neutral, 0: Strong Sell (Bearish).

- **Market Trend:** Stock indices momentum. (High: Upward trend).
- **Risk Environment:** VIX & Safe assets. (High Score: Low VIX/Stable Risk-on).
- **Economic Growth:** Employment, Production. (High: Expansion).
- **Interest Rates:** Levels & Direction. (High Score: Accommodative/Falling).
- **Inflation:** Prices & Commodities. (High Score: Stable/Disinflation. Low Score: Stagflation/Deflation).

Output Format (JSON only):

```
{
  "metrics": { "market_trend": { "label": "...", "score": 0-100}, ... },
  "summary": "<Implications for active management (approx 200 chars)>"
}
```

Macro Indicators [Format: Value (RoC: Value%)]:

- [1. **Rates & Policy**]
US Fed Rate: <val> (RoC: <val>) US 10Y Yield: <val> (RoC: <val>)
JP Policy Rate: <val> (RoC: <val>) JP 10Y Yield: <val> (RoC: <val>)
- [2. **Inflation & Commodities**]
US CPI: <val> (RoC: <val>) JP CPI: <val> (RoC: <val>)
Gold: <val> (RoC: <val>) Crude Oil: <val> (RoC: <val>)
- [3. **Growth & Economy**]
US Payrolls: <val> (RoC: <val>) Ind. Prod: <val> (RoC: <val>)
Housing Starts: <val> (RoC: <val>) Unemp. Rate: <val> (RoC: <val>)
JP Business Index: <val> (RoC: <val>)
- [4. **Market & Risk**]
USD/JPY: <val> (RoC: <val>) Nikkei 225: <val> (RoC: <val>)
S&P 500: <val> (RoC: <val>) US VIX: <val> (RoC: <val>)
Nikkei VI: <val> (RoC: <val>)

B.7 PM Agent

B.7.1 System Prompt.

System Prompt (Role Definition):

Role: You are the Chief Portfolio Manager (PM). Your task is to determine the definitive investment score by integrating **Top-Down Macro analysis** with **Bottom-Up Sector/Stock analysis**.

Decision Logic (Integration Strategy):

- **Goal:** Maximize alpha (1-month horizon) while strictly managing risk.
- **Harmonization:** Balance "Market Context" (Macro) vs. "Stock Specifics".
- **Macro Alignment:** If Macro is "Risk-Off", apply a conservative discount (lower scores) unless the stock has exceptional defensive qualities.
- **Tie-Breaking:** Use Macro context to resolve conflicts between Technicals and Fundamentals (e.g., High Inflation → favor fundamental pricing power over momentum).

Output Requirements:

- **Final Score:** 0-100 (Definitive conviction).
- **Rationale:** Decisive professional rationale justifying the integration.

B.7.2 User Prompt.

Instruction: As the Portfolio Manager, review the Macro and Sector-level inputs to provide your final investment decision for the next 1 month.

Input Reports (Integration):

- [1. **Macro Environment Report**]
<Macro Analyst Output (JSON)>
- [2. **Sector Specialist Report**]
<Sector Specialist Output (JSON)>

Final Decision Tasks:

1. **Final Investment Score (0-100):**
 - 100: Maximum Overweight (High conviction, perfect alignment).
 - 50: Neutral/Hold (Market-weight, no clear edge).

· 0: Strong Underweight/Avoid (High risk).

2. Final Rationale (Japanese):

- Explain macro influence on this specific stock.
- Summarize key reasons & risk-reward balance (30 days).

Output Format (JSON only):

```
{
  "final_score": <integer 0-100>,
  "reason": "<Decisive rationale in Japanese (150-200 chars)>"
}
```

C Performance by Portfolio Optimization

Table 4 shows the detailed performance by optimizing the portfolios constructed with the agents' outputs and index (TOPIX 100). In the table, return and volatility are annualized, and the Sharpe Ratio assumes a risk-free rate of 0%. Agent Strategies refers to the Risk Parity portfolio combining six LLM-based agent strategies. Transaction cost is applied one-way at 10bps per trade for Agent Strategies.

Table 4: Performance Comparison

Portfolio	Gross			Net (10bps)		
	Ret.	Vol.	S.R.	Ret.	Vol.	S.R.
TOPIX 100	19.3%	11.5%	1.68	19.3%	11.5%	1.68
Agent Strategies	13.7%	11.2%	1.22	10.6%	11.2%	0.95
50-50 Combined	16.8%	8.0%	2.11	15.2%	8.0%	1.91

D Representative Words in Outputs of Agents

Regarding the log-odds analysis, we first aggregate all text outputs and perform morphological analysis on them. We use the Japanese morphological analyzer MeCab [23] and the dictionary NEologd [56]. From the texts, we extracted only nouns without basic stopwords, created a corpus by fine-grained and coarse-grained settings, and applied log-odds analysis. Table 5 shows the top 10 words for each group. The words in the table are originally in Japanese but are shown translated into English.

Table 5: Top 10 representative words by Log-Odds Ratio with Dirichlet Prior

Agent	Fine-grained	Coarse-grained
Technical (Level 1)	Momentum, Neutral, Short-term, Favorable, Implication, Suggestion, Long-term, Condition, Decline, Volatility	Price, Trend, Rise, Upward-trend, Fall, Stock-price, Recent, Recommend, Continuation, Expectation
Quant. (Level 1)	Soundness, Margins, Growth, Profitability, Growth-rate, Favorable, Deterioration, Financials, Issues, Unstable	EPS, Stability, Improvement, Profit, Earnings, Trend, Stagnation, Increase, Attractive, Stability
Sector (Level 2)	Momentum, Soundness, Financials, Margins, Profitability, Growth-rate, Indicators, ROE, Efficiency, Undervalued	Overall, Trend, EPS, Improvement, Operating-profit, Stagnation, Net-income, Stock-price, Trend, Decrease
PM (Level 3)	Momentum, Soundness, Financials, Profitability, Undervalued, Issues, Undervaluation, Margins, Overvaluation, Attractive	Sector, Trend, Improvement, Stagnation, EPS, Average, Stability, Decrease, Target, Downward