KNOWING WHEN TO ABSTAIN: RISK-AWARE CRITERIA FOR ACTION-LEVEL SELECTIVITY IN TRADING AGENTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Despite continuous performance improvements of large language models, unreliable outputs may still be generated due to the inherent uncertainty of the models. For LLM-based trading agents, unreliable and useful actions may coexist within trading recommendations, which can lead to poor portfolio returns and elevated risks. One solution is to abandon those high-risk trading actions. For this purpose, Risk-Aware Action Selective Execution Criterion (RAASEC) is proposed. Three metrics are adopted to evaluate the impact of model uncertainty on each action, and two additional metrics are adopted to assess the market risk. The criterion ensures that only low-risk actions are selected for execution via an adaptive risk–coverage balancing strategy. Backtesting on the challenging China A-share market demonstrates that endowing agents with the ability to abstain from high-risk actions effectively enhances portfolio returns and outperforms the sort-of-states methods. Experiments also verify the complementarity of the five metrics.

1 INTRODUCTION

Despite the remarkable performance of large language models (LLMs) in market sentiment analysis Xiao (2024), trading strategy mining Chen & Kawashima (2026), and market trend forecasting Yu et al. (2023), LLM-powered autonomous trading agents achieve unsatisfactory performance on benchmarks such as AlphaForgeBench Zhang et al. (2026), AI-trader Fan et al. (2025) and Alpha Arena Alpha Arena (2025). It is observed that the agents may generate multiple trading action recommendations under identical market conditions, yet these actions are not equally reliable. Even a small number of unreliable actions blended with sensible recommendations can substantially erode portfolio returns once they are executed. The inherent uncertainty of LLMs is the core reason they generate unreliable recommendations, exposing trading agents to avoidable risks. Smaller models exhibit more serious defects of this kind.

Some existing studies adopt multi-agent architectures to feed multi-dimensional, fine-grained signals into LLMs Yu et al. (2024); Xiao et al. (2025); Ding et al. (2025), while others embed reflection mechanisms inside trading agents to enable self-evolution Li et al. (2024); Wang et al. (2024). Although both paradigms enhance agents' decision-making capabilities and do yield higher-quality trading recommendations, unreliable actions remain prevalent, leaving a fundamental question unaddressed: which candidate actions should be withheld from execution. A risk management module is adopted in TradingAgents Xiao et al. (2025) to adjust the trading recommendations for risk control purposes, which seemingly addresses the aforementioned question. However, the adjusted outputs still originate from LLMs, the question needs to be answered again.

How can agents determine whether an individual LLM-generated trading action is worth executing? A viable solution lies in quantifying the models' uncertainty, for which LLM confidence serves as a widely adopted metric. Existing confidence scoring methods include white-box approaches Kadavath et al. (2022); Gupta et al. (2024) tailored for opensource LLMs, black-box approaches Vashurin et al. (2025); Tian et al. (2023) applicable to closed-source commercial LLM APIs, and a variety of confidence calibration strategies built upon self-consistency Xiong et al. (2024) or perplexity Zhou et al. (2025). However most of them are designed to quantify uncertainty over the complete model response. Action-level confidence and corresponding calibration strategies remain underexplored.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Confidence scores capture an LLM's self-certainty regarding its generated predictions. Nevertheless, when LLMs exhibit overconfidence in unreliable recommendations, such scores fail to measure potential trading risks. To address this limitation, market risk factors need to be integrated for enhanced risk control. LLM-as-Judge offers a viable paradigm to assess market risks associated with specific trading actions Wang et al. (2024), while alternative methods merit further consideration.

Risk-based trading action abstention constitutes a risk-coverage trade-off. The objective is twofold: cutting trading risks while avoiding excessive action abstention, which essentially seeks an optimal balance between the two metrics to maximize investment returns. Since the agent's recommendations and corresponding risk values fluctuate across all decisions, static risk or coverage thresholds are difficult to predefine. Instead, such thresholds need to be dynamically calibrated according to real-time decision scenarios.

We propose **Risk-Aware Action Selective Execution Criterion (RAASEC)** , which evaluates each candidate action in terms of both model-endogenous risks and market risks. Actions satisfying calibrated criterion are executed, while the remaining actions are withheld. Our main contributions are summarized as follows:

(i) We propose measuring the risk of an individual trading action from two complementary perspectives: decision stability across reasoning views, which corresponds to model-endogenous risk and is aggregated from multiple action-level confidences; and execution feasibility under market and portfolio evidence, which aligns with market risk and is evaluated based on a LLM based evaluator and historical data.

(ii) We propose a risk-coverage adaptive balancing strategy to dynamically select actions for execution.

(iii) We verify that action-level selective execution serves as an effective mechanism to mitigate the impacts of model uncertainty and boost the performance of trading agents.

We evaluate RAASEC at two levels: first, whether its scores predict action-level validity; second, whether selective execution improves trading performance. All experiments are conducted on historical data from China's A-share market, which is viewed as more challenging for autonomous trading agents compared with other markets Fan et al. (2025). Those backtesting experiments closely simulate real-time trading environments and thoroughly eliminate look-ahead bias and test set leakage. A suite of comprehensive evaluation metrics is adopted to measure trading agent performance, including annualized return (AR), Sharpe ratio (SR), maximum drawdown (MDD), and information ratio (IR).

2 Related Work

2.1 Confidence Estimation for LLMs

Confidence estimation schemes for LLMs can be categorized into two primary types. (i) White-box methods, which derive confidence from token-level probabilitiesGupta et al. (2024), or proxy probing models trained on internal hidden states Kadavath et al. (2022); Chen et al. (2026).These approaches exhibit strong model dependency. (ii) Black-box methods, which elicit verbalized confidence directly from LLMs Xiong et al. (2024); Tian et al. (2023), or assess the semantic agreement across multiple sampled outputs Vashurin et al. (2025). Such approaches are model-agnostic. Furthermore, existing research also demonstrates that combining black-box and white-box approaches can yield more accurate confidence Vashurin et al. (2025). Verbalized confidence generally requires consistency-based calibration Xiong et al. (2024); Tian et al. (2023); Zhou et al. (2026), prompting LLMs to generate multiple responses and then aggregating their confidence scores to yield a more reliable one. Calibration can also be achieved via LLM fine-tuning Xu et al. (2024); Li et al. (2026), albeit with extra training data. All above confidence scores are response-level.

2.2 Risk-based Selection in Financial Agents

Kou et al. (2025) selects alpha according to their confidence scores, retaining only those quantitative strategies that satisfy a predefined confidence threshold. Dimino et al. (2025) reveals the relationship

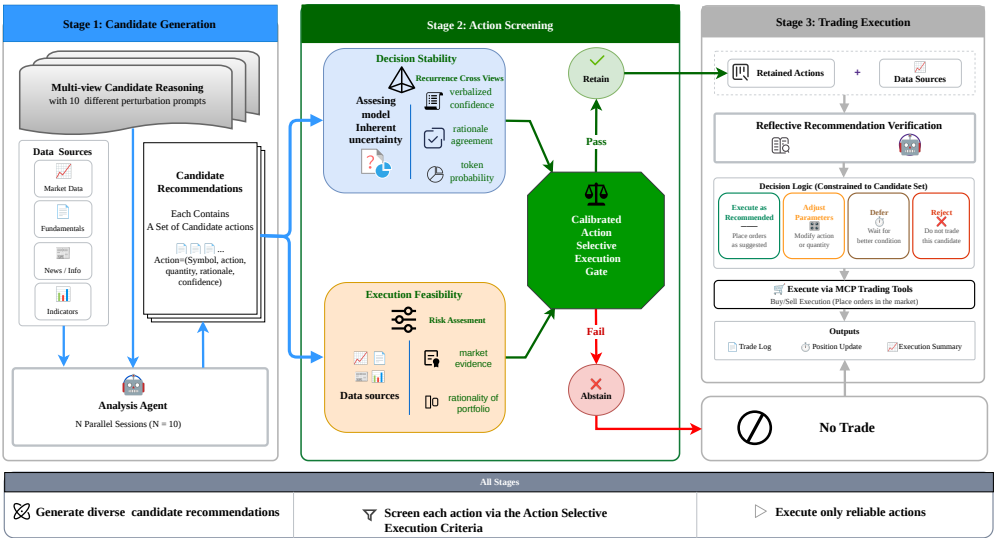


Figure 1: Overview of the proposed three-agent serial architecture. The three agents operate sequentially: the Analysis Agent generates candidate trading recommendations, the Confidence Filtering Agent ranks and filters them according to confidence, and the Trading Agent verifies them through reflective reasoning before execution.

between firm-level confidence and risk factors. TraderExpert Ding et al. (2025) adopts an LLM as the comparator to conduct pairwise comparisons across all stocks based on their earnings prospects, ranked by win counts, and the top-k candidates are selected for trading. It is worth pointing out that such pairwise-comparison-based selection schemes incur an $O(n^2)$ computational complexity and suffers from non-transitivity. Multi-agent trading frameworks such as TradingAgents Xiao et al. (2025) and QuantAgent Wang et al. (2024) deploy independent risk sub-agents to audit candidate trading actions, yet their risk judgment still relies on LLM reasoning, introducing extra uncertainty. TiMi Song et al. (2026) adopt static hard thresholds for drawdown and exposure limits, failing to adequately assess the model's inherent uncertainty and dynamically balance the risk-coverage trade-offs.

3 Selective Execution Action Framework

3.1 Overview of Action-Selective Trading

At each trading time t, the base LLM produces a set of candidate actions rather than a single final decision. Each candidate action may be plausible, yet not every proposed action should be executed. We formulate trading as an action-selective decision process that distinguishes action generation from action execution.

The framework contains three stages is shown in figure 1. First, a candidate generator produces a diverse set of stock-action recommendations from the observed market state. Second, an action screening module evaluates each candidate according to the proposed action selective execution criteria and returns an executable subset. Third, a fixed execution module verifies the actions in the subset. The key design is the screening stage: it determines whether each individual stock-action pair should be executed or abstained from. The execution criterion combines decision stability and execution feasibility, which is formalized in Section 4.

3.2 Candidate Action Generation

We separate candidate generation from execution: an action can be proposed by the LLM but should be executed only after passing the proposed execution criterion.

Under a general ReAct paradigm, the agent performs market data analysis, fundamental analysis, technical indicator analysis, and news sentiment analysis, generating candidate trading recommendations through a set of perturbation prompts. Following comparative performance evaluations,

Qwen3-32B Yang et al. (2025) is selected as the base model. The LLM is deployed locally and autonomously invokes various tools to collect market information.

Thirteen widely adopted technical indicators and company fundamental information are retrieved from remote MCP servers. Market data including prices and trading volumes is obtained locally. News articles, corporate disclosures, and institutional research reports are sourced from a pre-crawled local text database. During retrieval, relevant documents corresponding to the target stock and trading date are extracted, with a tunable parameter controlling the number of retrieved items. Strict temporal filtering is applied during backtesting to prevent look-ahead bias.

As an output generated by the LLM, the trading recommendation $Y = \{y_1, y_2, \ldots, y_N\}$ consists of multiple action recommendations y_n. The value of N is determined by the LLM and satisfies the constraint $N \leq N_{\max}$. Each action recommendation y_n is defined as a 5-tuple $(s_n, a_n, q_n, r_n, c_n)$, where s_n denotes the stock ticker symbol; a_n represents the trading action; q_n indicates the trading quantity, which must be an integer multiple of 100 shares to comply with the trading rules of China's A-share market; r_n refers to the decision rationale for recommending the corresponding action , which is expressed in natural language and corresponds to LLM-adopted quantative strategies. c_n denotes the verbalized confidence generated by the LLM.

Following the idea of SteerConf Zhou et al. (2026), the analysis agent is prompted with multiple sentiment-based perturbation templates to generate multiple recommendations, while keeping the underlying market information unchanged. Each contains a set of actions, yielding a pool of candidate actions. Numerous unreliable actions are mixed within this pool and require screening. The use of multiple candidate recommendations introduces additional computational overhead, resulting in increased computation time. However, in this setting, since only one trade is executed per day, the sensitivity to timing is not particularly strong.

3.3 Action Selective Screening and Execution

At the action screening stage, each candidate action is evaluated for decision stability (model confidence) and execution feasibility (market risk). These are integrated into a unified score, and an adaptive threshold selects higher-scoring actions (lower risk), permitting execution only when both constraints are satisfied. This is termed the Risk-Aware Action Selective Execution Criterion (Section 4).

At the trading execution stage, an agent using the same base model and market information verifies the screened actions via risk-guided reflection within one trading cycle, then calls the MCP trading tool to execute buys and sells.

4 Risk-Aware Action Selective Execution Criterion

Under RAASEC, a candidate action is worthy of execution only when it satisfies the following two complementary conditions: it should remain stable across alternative reasoning views, and it should be feasible under external market and portfolio evidence. Given M trading recommendations $Y_1, Y_2, \ldots, Y_M$ generated by the analysis agent based on prompts of different forms. The i-th response consists of N_i actions, denoted as $Y_i = \{y_1^i, y_2^i, \ldots, y_{N_i}^i\}$. All actions in those recommendations constitute an action set $\mathcal{A}$. Each element y in the set $\mathcal{A}$ is a 5-tuple (s, a, q, c, r). For convenience, an indicator function is defined to determine whether two actions hold the identical stock-action pair, formulated as:

$$\mathbf{1}(y_i, y_j) = \begin{cases} 1 & \text{if } s_i = s_j \text{ and } a_i = a_j \\ 0 & \text{otherwise} \end{cases}. \tag{1}$$

4.1 Decision Stability

The decision stability of an action can be characterized by the LLM confidence associated with the action. By aggregating diverse responses under multiple reasoning views, three action-level confidence metrics derived from different strategies are adopted to evaluate the decision stability.

4.1.1 REGULARIZED VERBALIZED CONFIDENCE

For action y, its regularized verbalized confidence (RVC) is calculated as follows:

$$\text{RVC}(y) = \frac{1}{M} \sum_{i=1}^{M} \sum_{j=1}^{N_i} c_j^i \cdot \mathbf{1}(y, y_j^i), \tag{2}$$

where c_j^i denotes the verbalized confidence of the j-th action in the i-th response. A higher RVC indicates that the action is repeatedly generated across perturbations and consistently associated with higher verbalized confidence. In implementation, each raw verbalized confidence in $[0, 100]$ is scaled to $[0, 1]$ before Eq.2, where $\mathbf{1}(y, y_j^i)$ follows Eq. 1, so $\text{RVC}(y) \in [0, 1]$.

4.1.2 REGULARIZED FREQUENCY BASED CONFIDENCE

Regularized frequency based confidence (RFBC) of action $y = (s, a, q, r, c)$ is calculated as follows:

$$\text{RFBC}(y) = \frac{1}{M} \sum_{i=1}^{M} \sum_{j=1}^{N_i} u(y, y_j^i) \tag{3}$$

where

$$u(y, y_j^i) = \begin{cases} \text{sim}(r, r_j^i), & \text{if } s = s_j^i,\ a = a_j^i, \\ p_1, & \text{if } s = s_j^i,\ a \leftrightarrow a_j^i, \\ p_2, & \text{if } s = s_j^i,\ a \not\leftrightarrow a_j^i, \\ 0, & \text{otherwise}, \end{cases} \tag{4}$$

p_1 and p_2 are two negative-valued penalty coefficients. The notation $a \leftrightarrow a_j^i$ denotes a and a_j^i are deemed starkly contradictory (one action is buy and the other is sell), whereas other pairs of distinct actions that are regarded as weaker contradictory, denoted by operation symbol $\not\leftrightarrow$. $\text{sim}(r, r_j^i)$ measures the agreement between two decision rationales r and r_j^i by the character-level bigram Jaccard similarity. RFBC captures the occurrence frequency of (s, a), which can be interpreted as the consensus across responses, yet penalizes contradictory actions and identical actions with inconsistent rationales. When RFBC is non-positive due to p_1/p_2, we replace it by a small $\varepsilon > 0$ before the geometric mean.

4.1.3 TOKEN PROBABILITY BASED CONFIDENCE

Token probability based confidence (TPBC) leverages internal signals of the LLM to measure the uncertainty of output segments corresponding to specific actions, making it inherently a white-box confidence method. For action y, suppose it corresponds to a continuous token sequence of length T within the LLM output, a token probability matrix $(p_{ti})_{T \times k}$ can be obtained when retaining merely the top-k probability of token classes at each time step. The uncertainty of the output at time step t can be quantified via local entropy H_t:

$$H_t = -\sum_{i=1}^{k} p_{ti} \log (p_{ti} + \varepsilon) \tag{5}$$

where ε is a small numerical constant. The uncertainty of the entire token sequence can be measured by the standard deviation σ_y of the log-likelihood values corresponding to all tokens in the sequence. By integrating these two uncertainty metrics, the confidence score of y can be computed as:

$$\text{conf}_{\text{TPBC}}(y) = e^{-\left(\sum_{t=1}^{T} H_t + \sigma_y / \tau\right)/2} \tag{6}$$

where τ is a temperature parameter applied to the standard deviation term to prevent its scale from overpenalizing the score. When calculating this confidence score, structural JSON tokens and fixed key names need to be excluded.

This confidence is calibrated on all the responses as:

$$\text{TPBC}(y) = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N_i}\omega_j^i \cdot \text{conf}_{\text{TPBC}}(y_j^i)}{\sum_{i=1}^{M}\sum_{j=1}^{N_i}\omega_j^i} \tag{7}$$

where function $\omega_j^i = T(y_j^i) \cdot \mathbf{1}(y, y_j^i)$ and $T(y_j^i)$ returns the length of the token sequence corresponding to y_j^i.

4.2 Execution Feasibility

Stability is necessary but insufficient. LLM may generate erroneous outputs due to knowledge gaps and reasoning errors, yet exhibit extreme overconfidence in their outputs. To address this issue, external information needs to be introduced to quantify the trading risks induced by each action. This paper adopts two action-level risk assessment approaches: the first evaluates the rationality of portfolios using historical data, and the second conducts a comprehensive assessment of the market risks of specific action via LLMs. For convenience, the scores obtained from these two assessment methods are also referred to as confidence in this work.

4.2.1 Intra-Portfolio Signal Mismatch Risk Based Confidence (IPSMRBC)

Intra-portfolio signal mismatch describes the phenomenon whereby return forecasts of distinct stocks within a portfolio are inconsistent or mutually contradictory. Intra-portfolio signal mismatch risk based confidence (IPSMRBC) can identify erroneous decisions even when the LLM maintains strong confidence in them. The degree of signal mismatch between two trading actions y_i and y_j can be measured by function b, as follows:

$$b(y_i, y_j) = \frac{1 + s(y_i, y_j) \cdot \rho(y_i, y_j)}{2} \tag{8}$$

where $\rho(y_i, y_j)$ returns the Spearman rank correlation between the trading prices of stock y_i and y_j over the 120 trading days preceding the current trading day. It measures the strength of monotonic association between two variables and ranges from -1 to $+1$. A positive value implies executing consistent actions on y_i and y_j can be regarded as reliable. Nevertheless, the impact of different directional actions on the two stocks should also be taken into account. A prior function $s(y_i, y_j)$ is adopted to map $\rho(y_i, y_j)$ to a reasonable confidence score. It takes a value of 1 when both actions are the same, -1 if one is sell and the other is buy, and some empirical values are predefined for pairs involving other different action directions.

Hence, the confidence score associated with specific action y can be calculated as follows:

$$\text{IPSMRBC}(y) = \frac{1}{\sum_{i=1}^{M} N_i}\sum_{i=1}^{M}\sum_{j=1}^{N_i} b(y, y_j^i) \tag{9}$$

4.2.2 Multi-evidence Risk Based Confidence

Multi-evidence Risk Based Confidence (MRBC) introduces an LLM to evaluate the (s, a, q, r) attributes of each candidate trading action and identify potential investment risks. Rather than introducing an additional active risk agent into the decision-making pipeline, akin to the practice of TradingAgents Xiao et al. (2025), MRBC only serves as an evaluator, conducting a post-hoc evidence verification to examine whether the proposed actions satisfy external feasibility constraints and to give a confidence score. Importantly, MRBC only operates on actions that have already entered the candidate set and does not generate, modify, or expand the action space.

Within this framework, the LLM constructs a four-dimensional risk radar to organize and evaluate multi-source risk evidence. From a Chief Risk Officer (CRO) perspective, the verifier performs a comprehensive assessment of latent risks associated with each candidate action and produces a risk-aware confidence score. Specifically, the verification process evaluates the execution feasibility of the proposed stock transactions and position sizing, while further examining whether the quantitative strategy signals and generated explanations from the analysis agent are consistent with external market conditions. During MRBC computation, the risk evaluation module functions as a passive confidence validation mechanism, providing additional risk evidence for confidence calibration without participating in the action generation process.

4.3 Adaptive Calibration and Selective Execution

Five confidence scores are derived via two distinct evaluation perspectives. Theoretically, a candidate trading action satisfies Risk-Aware Action Selective Execution Criterion (RAASEC) and is eligible for inclusion in the executable list only if all five confidence values exceed their respective predefined thresholds. Nevertheless, deploying five separate thresholds in practical industrial applications introduces excessive complexity. To address this issue, the geometric mean of the five confidence scores is computed to aggregate these metrics into a unified single confidence score. In implementation, actions are screened by this aggregated score together with a coverage threshold, rather than by five separate hard cuts.

Subsequent analyses reveal that this aggregated confidence score fluctuates drastically across individual decision-making steps, rendering it infeasible to calibrate a reasonable risk threshold during the model training phase. The problem of screening valid trading actions essentially constitutes a risk-coverage trade-off optimization problem with return rate as the objective function: a lower risk threshold corresponds to higher coverage (i.e., a larger selection ratio). Accordingly, a coverage threshold is adopted as the screening criterion instead. Given that the number of trading actions recommended by the LLM is fixed per decision round, the coverage threshold provides a direct and interpretable control over the selection ratio. Furthermore, a threshold calibration paradigm is adopted, wherein thresholds are finalized on the validation set and subsequently deployed on the test set, which substantially enhances practical implementability.

Let $O = |\mathcal{A}|$ denote the total number of recommended actions in each decision round. Since O is fixed, the coverage threshold directly determines how many actions are selected for execution. On this basis, the proposed selective execution algorithm proceeds as follows: (i) calculate the geometric mean of the five confidence scores for each action in $\mathcal{A}$ and rank all actions by this aggregated value; (ii) determine the selection quantity $k = \max(1, \lceil TH \times O \rceil)$, where TH is the coverage threshold and $O = |\mathcal{A}|$; (iii) conduct a grid search over the adaptive calibrated coverage threshold on the validation set and select the optimal TH against the return; and (iv) during testing, compute k from the predefined TH and O, then adopt the Top-k strategy to populate the executable action list.

5 Experiment

5.1 Experimental Setup

We evaluate our framework on the constituent stocks of the CSI 300 Index, a representative benchmark of the Chinese equity market. Historical price data are collected from Baostock library with a missing rate below 1%. The backtesting period spans from January 1, 2025 to December 16, 2025. Data corresponding to the January 1, 2024–April 1, 2024 window constitute the validation set, on which hyperparameter tuning is performed. These hyperparameters are subsequently fixed for the subsequent testing process. We choose Chinese A-shares as the testbed because existing trading agents often struggle to achieve robust performance in this challenging market.

Several models, including GPT-4o, Claude-3.5, Gemini, Llama-3.1-70B, DeepSeek-V3, and Qwen variants, are evaluated individually for this task. Qwen3-32B was selected for superior Chinese market understanding, reproducibility brought by local deployment and flexibility in real-world implementation. The model is deployed locally via SGLang with tensor parallelism and runs the trading agent through an OpenAI-compatible endpoint using Python 3.12 and PyTorch 2.9. Agent decoding uses the SGLang default sampling temperature (T=1.0). The relatively small model size indeed creates a more challenging experimental setting. All experiments run on a Linux server (1 TB RAM, eight RTX 4090 24 GB GPUs).

5.2 Action-Level Metrics Validation

We validate the five confidence metrics under the assumption that sell actions are valid if subsequent prices decline, and buy/hold actions are valid if they rise. Although not perfectly rigorous, this assumption helps quantify practical effectiveness. For each decision, the average price over the next five trading days is compared with the current price, and valid days must account for at least 50% of samples. Since RAASEC performs top-k selection, actions are sorted by confidence and ordinal ranks are used for evaluation. Table 1 shows that all metrics achieve a valid rate around 80%, with

Table 1: Validity Verification Results of Action-Level Confidence metrics for CSI300.

Score	Valid Rate	AUROC↑	AUPRC↑	ECE↓
RVC	86.5%	0.569	0.870	0.102
RFBC	**87.6%**	0.552	0.900	0.079
TPBC	76.5%	0.644	0.822	0.161
IPSMRBC	81.0%	**0.704**	**0.924**	**0.067**
MRBC	75.0%	0.583	0.783	0.210

Table 2: End-to-end trading performance comparison. ↑ indicates higher is better, while ↓ indicates lower is better. Active Days denotes the number of trading days on which at least one trade is executed.

Method	AR↑	Alpha↑	β	SR↑	IR↑	MDD↓	Active Days
Buy and Hold	10.48%	2.96%	0.3770	0.5171	-0.42	10.59%	0 (0%)
Only Analysis	17.22%	14.28%	0.0365	1.1791	-0.1058	**7.29%**	147 (63.36%)
Analysis+Verify	26.69%	19.19%	0.2400	1.4812	0.3152	13.50%	166 (71.55%)
TradingAgents	41.05%	32.88%	0.0751	2.0800	0.7400	11.09%	183 (78.88%)
FinAgent	13.86%	5.54%	0.4272	0.5700	-0.2200	22.43%	163 (69.66%)
RAASEC	**52.57%**	**36.34%**	0.4196	**2.1149**	**1.0715**	16.83%	92 (39.66%)

RFBC highest (87.6%) and MRBC lowest (75%). No single metric is reliable enough for independent screening, but their complementary validity judgments support integration into RAASEC. Although IPSMRBC attains the best AUROC, AUPRC, and ECE, its 81% valid rate and mediocre ranking performance (AUROC ≈ 0.7) prevent standalone use.

We compare RAASEC with three baselines and two representative methods: FinAgent Zhang et al. (2024), a single-agent framework with memory-driven reflection and experience retrieval, and TradingAgents Xiao et al. (2025), a multi-agent paradigm with role collaboration and debate. Both of them are based on Qwen3-32B in this experiment. As shown in Table 2, Buy and Hold achieves 10.48% AR and 0.52 SR. Only Analysis that fulfills only the first stage improves all metrics (17.22% AR), and Analysis+Verify that lacks the second stage further raises AR to 26.69%, SR to 1.48, and IR to 0.32, validating reflective verification. RAASEC achieves the best overall performance: 52.57% AR, 36.34% alpha, 2.11 SR, and 1.07 IR, despite a 16.83% MDD. It trades on only 92 of 232 days (39.66%), indicating that on most decision dates the model predominantly adopts *hold* and *watch* rather than executing trades, which demonstrates that the proposed confidence scores effectively filter out unreliable and high-risk recommendations. The gain stems from both confidence scores and multi-sampled candidates, but the larger pool adds noise and risk, necessitating RAASEC. TradingAgents reaches 41.05% AR, 32.88% alpha, and 2.08 SR, competitive but below RAASEC. FinAgent yields only 13.86% AR, 0.57 SR, and -0.22 IR, underperforming even Only Analysis. The table reports the best run; other runs' AR deviates within 2%.

Figure 2 shows that AR, alpha, SR, and IR all of RAASEC peak at $TH = 30\%$. Lower thresholds admit too many low-confidence trades, while higher thresholds are overly conservative and miss profitable opportunities. Although MDD increases moderately, $TH = 30\%$ achieves the best return-risk trade-off.

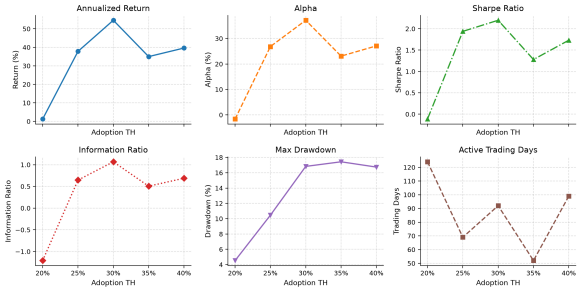


Figure 2: Sensitivity analysis of the adoption threshold TH. Annualized return, alpha, Sharpe ratio, and information ratio all peak at $TH = 30\%$, while maximum drawdown increases with TH up to 35% and the number of active trading days is minimized at $TH = 35\%$.

Table 3: Ablation Results of Adaptive Coverage Threshold.

Method	AR↑	Alpha↑	β	SR↑	IR↑	MDD↓	Active Days
RAASEC without RFBC	44.81%	31.18%	0.4864	1.4998	0.7824	16.91%	102 (43.97%)
RAASEC without RVC	42.79%	28.76%	0.4490	1.8025	0.8373	13.81%	61 (26.29%)
RAASEC with fixed coverage threshold	30.12%	20.91%	0.3027	1.6434	0.4630	11.63%	63 (27.16%)
RAASEC with Top-k Filtering	27.17%	16.35%	0.4688	1.2488	0.3573	16.80%	114 (49.14%)

Table 4: Ablation study on confidence score components .

Method	τ	AR↑	Alpha↑	β	SR↑	IR↑	MDD↓	Active Days
RAASEC (full)	30%	**52.57%**	**36.34%**	0.4196	**2.1149**	**1.0715**	16.83%	92 (39.66%)
w/o RVC	25%	43.50%	30.74%	0.4104	1.6158	0.7820	15.60%	39 (16.81%)
w/o IPSMRBC	25%	21.60%	16.46%	0.2691	1.1747	0.0662	11.09%	129 (55.60%)
w/o TPBC	30%	49.46%	35.41%	0.3609	2.0817	1.0512	13.64%	62 (26.72%)
w/o MRBC	25%	42.46%	29.03%	0.4409	1.7061	0.8034	17.28%	58 (25.00%)
w/o RFBC	30%	32.24%	26.55%	0.0712	1.7463	0.4834	**6.49%**	123 (53.02%)

5.3 Ablation Experiments on Adapative Coverage Threshold

A set of ablation experiments also validate the efficacy of the adaptive coverage threshold. As listed in Table 3, the approach equipped with a fixed coverage threshold only achieves an annualized return rate of 30.12%, considerably lower than the 52.57% yielded by the adaptive coverage threshold scheme. Furthermore, the Top-k method with a fixed k value reaches a maximum annualized return of 27.17%, which is inferior to the approaches relying on the coverage threshold mechanism. It is demonstrated that the coverage threshold serves as an effective technique to strike a balance between risk and coverage. In addition, the adaptive adjustment of the coverage threshold driven by overall confidence contributes to improved adaptability amid sophisticated decision-making scenarios.

Within the RAASEC framework, overall confidence integrates both RVC and RFBC metrics. Experimental data in Table 3 reveals that standalone utilization of either indicator leads to degraded performance. The fusion of information from heterogeneous perspectives enables more precise quantification of model uncertainty.

5.4 Ablation Study On Five Criteria

As shown in Table 4, we remove one confidence component at a time from the full RAASEC framework to assess its individual contribution. Full RAASEC achieves the best overall performance, with 52.57% AR, 36.34% alpha, 2.11 SR, and 1.07 IR. Removing IPSMRBC causes the most severe degradation, dropping AR to 21.60% and IR to 0.07, which indicates that intra-portfolio signal mismatch risk is critical for filtering out inconsistent trading actions. This result suggests that without IPSMRBC, the framework loses its ability to detect contradictory signals across correlated stocks, leading to excessive exposure to unreliable trades.

Removing RFBC yields 32.24% AR but achieves the lowest MDD of 6.49%, revealing a trade-off between return and drawdown control. Removing RVC, MRBC, or TPBC leads to moderate declines, with AR of 43.50%, 42.46%, and 49.46%, respectively. Among them, TPBC has the smallest impact, yet the full RAASEC still outperforms it on all major metrics, highlighting the complementary role of token-level uncertainty. Overall, these results confirm that the five metrics capture distinct aspects of action reliability, and their fusion is essential for effective risk-coverage balancing.

6 Conclusion

In this work, Action Selective Execution Criteria are utilized to enhance decision-making capacity of our trading agent. Five action-level risk assessment metrics are proposed considering decision stability and execution feasibility, which together constitute RAASEC. Only candidate actions meeting the criterion can be executed, and the rest are filtered out due to perceived potential risks. This abstention mechanism offers a feasible risk control strategy against unreliable decisions caused by the inherent uncertainty of LLMs. Backtests on the A-share market verify that RAASEC substantially improves the profitability of the stock trading agents

AI USE STATEMENT

In this work, we used generative AI tools to aid or polish writing, including improving language clarity, grammar, and overall presentation. We have not used generative AI tools for research ideation or execution, drafting sections of the paper, generating synthetic datasets, or proving mathematical claims. Additionally, we used generative AI tools for retrieval and discovery, including finding potentially relevant literature and related work. We have reviewed all AI-assisted work. AI-assisted writing was reviewed by the authors for factual accuracy, clarity, and consistency, and literature identified through AI-assisted retrieval was checked against the original sources. We take responsibility for the final content of this work, including text, claims or artifacts produced with the aid of generative AI.

REPRODUCIBILITY STATEMENT

To facilitate the reproducibility of our results, we will release the source code and datasets used in our experiments upon acceptance. The released repository will include the implementation of the proposed RAASEC framework, experimental configurations, data processing scripts, and instructions for reproducing the reported results. Detailed descriptions of the methodology, experimental settings, and ablation studies are provided in the main paper and appendix. These materials will allow researchers to reproduce our experiments and further evaluate the proposed method.

REFERENCES

Alpha Arena. Ai trading in real markets. https://nof1.ai/, 2025. Accessed January 23, 2026.

Qizhao Chen and Hiroaki Kawashima. Adaptive alpha weighting with ppo: enhancing prompt-based llm-generated alphas in quant trading. *International Journal of Data Science and Analytics*, 22, 03 2026. doi: 10.1007/s41060-026-01072-2.

Zijun Chen, Wenbo Hu, and Richang Hong. Deep hidden cognition facilitates reliable chain-of-thought reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 37295–37304, 2026. doi: 10.1609/aaai.v40i44.41061.

Fabrizio Dimino, Krati Saxena, Bhaskarjit Sarmah, and Stefano Pasquali. Uncovering representation bias for investment decisions in open-source large language models, 2025. URL https://arxiv.org/abs/2510.05702.

Qianggang Ding, Haochen Shi, Jiadong Guo, and Bang Liu. Tradexpert: Revolutionizing trading with mixture of expert llms, 2025. URL https://arxiv.org/abs/2411.00782.

Tianyu Fan, Yuhao Yang, Yangqin Jiang, Yifei Zhang, Yuxuan Chen, and Chao Huang. Ai-trader: Benchmarking autonomous agents in real-time financial markets, 2025. URL https://arxiv.org/abs/2512.10971.

Neha Gupta, Harikrishna Narasimhan, Wittawat Jitkrittum, Ankit Singh Rawat, Aditya Krishna Menon, and Sanjiv Kumar. Language model cascades: Token-level uncertainty and beyond, 2024. URL https://arxiv.org/abs/2404.10136.

S. Kadavath, T. Conerly, A. Askell, T. Henighan, D. Drain, E. Perez, N. Schiefer, Z. Hatfield-Dodds, N. DasSarma, E. Tran-Johnson, S. Johnston, S. El-Showk, A. Jones, N. Elhage, T. Hume, A. Chen, Y. Bai, S. Bowman, S. Fort, D. Ganguli, D. Hernandez, J. Jacobson, J. Kernion, S. Kravec, L. Lovitt, K. Ndousse, C. Olsson, S. Ringer, D. Amodei, T. Brown, J. Clark, N. Joseph, B. Mann, S. McCandlish, C. Olah, and J. Kaplan. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022. URL https://arxiv.org/abs/2207.05221.

Zhizhuo Kou, Holam Yu, Junyu Luo, Jingshu Peng, Xujia Li, Chengzhong Liu, Juntao Dai, Lei Chen, Sirui Han, and Yike Guo. Automate strategy finding with LLM in quant investment. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 18517–18533, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7.

doi: 10.18653/v1/2025.findings-emnlp.1005. URL https://aclanthology.org/2025.findings-emnlp.1005/.

Haohang Li, Yangyang Yu, Zhi Chen, Yuechen Jiang, Yang Li, Denghui Zhang, Rong Liu, Jordan W. Suchow, and Khaldoun Khashanah. Finmem: A performance-enhanced LLM trading agent with layered memory and character design. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*, 2024. URL https://openreview.net/forum?id=sstfVOwbiG.

Yibo Li, Miao Xiong, Jiaying Wu, and Bryan Hooi. Conftuner: Training large language models to express their confidence verbally. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=VZQ04Ojhu5.

Zifan Song, Kaitao Song, Guosheng Hu, Ding Qi, Junyao Gao, Xiaohua Wang, Dongsheng Li, and Cairong Zhao. Trade in minutes! rationality-driven agentic system for quantitative financial trading. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ROEwZAxqyS.

Katherine Tian, Eric Mitchell, Allan Zhou, Archit Sharma, Rafael Rafailov, Huaxiu Yao, Chelsea Finn, and Christopher Manning. Just ask for calibration: Strategies for eliciting calibrated confidence scores from language models fine-tuned with human feedback. In Houda Bouamor, Juan Pino, and Kalika Bali (eds.), *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 5433–5442, Singapore, December 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.emnlp-main.330. URL https://aclanthology.org/2023.emnlp-main.330/.

Roman Vashurin, Maiya Goloburda, Albina Ilina, Aleksandr Rubashevskii, Preslav Nakov, Artem Shelmanov, and Maxim Panov. Cocoa: A minimum bayes risk framework bridging confidence and consistency for uncertainty quantification in llms. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 106236–106281. Curran Associates, Inc., 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/98d2006bf15f00e2ff293dfba4acb8b2-Paper-Conference.pdf.

Saizhuo Wang, Hang Yuan, Lionel M. Ni, and Jian Guo. Quantagent: Seeking holy grail in trading by self-improving large language model, 2024. URL https://arxiv.org/abs/2402.03755.

Weizhen Xiao. A comparative review of advanced techniques for financial sentiment analysis. In *Proceedings of the International Conference on Modeling, Natural Language Processing and Machine Learning*, CMNM '24, pp. 76–80, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400709760. doi: 10.1145/3677779.3677791. URL https://doi.org/10.1145/3677779.3677791.

Yijia Xiao, Edward Sun, Di Luo, and Wei Wang. Tradingagents: Multi-agents llm financial trading framework, 2025. URL https://arxiv.org/abs/2412.20138.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, YIFEI LI, Jie Fu, Junxian He, and Bryan Hooi. Can LLMs express their uncertainty? an empirical evaluation of confidence elicitation in LLMs. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=gjeQKFxFpZ.

Tianyang Xu, Shujin Wu, Shizhe Diao, Xiaoze Liu, Xingyao Wang, Yangyi Chen, and Jing Gao. SaySelf: Teaching LLMs to express confidence with self-reflective rationales. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 5985–5998, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.343. URL https://aclanthology.org/2024.emnlp-main.343/.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jing Zhou, Jingren Zhou, Junyang Lin, Kai Dang, Keqin Bao, Kexin Yang,

Le Yu, Lianghao Deng, Mei Li, Mingfeng Xue, Mingze Li, Pei Zhang, Peng Wang, Qin Zhu, Rui Men, Ruize Gao, Shixuan Liu, Shuang Luo, Tianhao Li, Tianyi Tang, Wenbiao Yin, Xingzhang Ren, Xinyu Wang, Xinyu Zhang, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yinger Zhang, Yu Wan, Yuqiong Liu, Zekun Wang, Zeyu Cui, Zhenru Zhang, Zhipeng Zhou, and Zihan Qiu. Qwen3 technical report, 2025. URL https://arxiv.org/abs/2505.09388.

Xinli Yu, Zheng Chen, Yuan Ling, Shujing Dong, Zongyi Liu, and Yanbin Lu. Temporal data meets llm – explainable financial time series forecasting. Papers 2306.11025, arXiv.org, Jun 2023. URL https://ideas.repec.org/p/arx/papers/2306.11025.html.

Yangyang Yu, Zhiyuan Yao, Haohang Li, Zhiyang Deng, Yuechen Jiang, Yupeng Cao, Zhi Chen, Jordan W. Suchow, Zhenyu Cui, Rong Liu, Zhaozhuo Xu, Denghui Zhang, Koduvayur Subbalakshmi, GUOJUN XIONG, Yueru He, Jimin Huang, Dong Li, and Qianqian Xie. Fincon: A synthesized LLM multi-agent system with conceptual verbal reinforcement for enhanced financial decision making. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=dG1HwKMYbC.

Wentao Zhang, Lingxuan Zhao, Haochong Xia, Shuo Sun, Jiaze Sun, Molei Qin, Xinyi Li, Yuqing Zhao, Yilei Zhao, Xinyu Cai, Longtao Zheng, Xinrun Wang, and Bo An. A multimodal foundation agent for financial trading: Tool-augmented, diversified, and generalist, 2024. URL https://arxiv.org/abs/2402.18485.

Wentao Zhang, Mingxuan Zhao, Jincheng Gao, Jieshun You, Huaiyu Jia, Yilei Zhao, Bo An, and Shuo Sun. Alphaforgebench: Benchmarking end-to-end trading strategy design with large language models. Papers 2602.18481, arXiv.org, Feb 2026. URL https://ideas.repec.org/p/arx/papers/2602.18481.html.

Zhi Zhou, Tan Yuhao, Zenan Li, Yuan Yao, Lan-Zhe Guo, Yu-Feng Li, and Xiaoxing Ma. A theoretical study on bridging internal probability and self-consistency for llm reasoning. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 87380–87413. Curran Associates, Inc., 2025. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/7e9afa9a02857bce4515247842471444-Paper-Conference.pdf.

Ziang Zhou, Tianyuan Jin, Jieming Shi, and Li Qing. Steerconf: Steering LLMs for confidence elicitation. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2026. URL https://openreview.net/forum?id=5sgK63Zshg.

APPENDIX

This appendix provides supplementary details. Its organization is as follows:

- A. Technical and Fundamental Analysis Tools
  - A.1 Technical Indicator Tool
  - A.2 Fundamental Analysis Tool
- B. Financial News Corpus Construction
  - B.1 News Collection Pipeline
  - B.2 Document Processing
  - B.3 Document Ranking and Retrieval
- C. Analysis Agent Prompt
- D. Trading Agent Prompt
- E. Perturbation Prompts for Confidence Sampling
- F. Character-level Bigram Jaccard Similarity
- G. Token Selection for TPBC Computation
- H. Computation of Spearman Rank Correlation

- H.1 Historical Price Data
- H.2 Construction of Return Series
- H.3 Spearman Rank Correlation
- H.4 Implementation Details
- H.5 Lookback Window

- I. Risk Assessment Agent Prompt
  - I.1 Role Definition
  - I.2 Risk Radar
  - I.3 Holistic Risk Assessment
  - I.4 Evidence Constraints
  - I.5 Output Format
  - I.6 Reproducibility

A Technical and Fundamental Analysis Tools

This appendix summarizes the implementation details of the tools available to the agent. The Technical Indicator Tool and the Fundamental Analysis Tool are implemented as MCP (Model Context Protocol) services and retrieve financial data from the Tushare Pro database. To improve efficiency and ensure reproducibility, all queried results are cached locally after the first request. Subsequent requests for the same stock and trading date are served directly from the local cache, ensuring identical inputs across repeated experiments.

The Market Data Tool and the News and Announcement Retrieval Tool operate entirely on locally stored datasets. During inference, all returned market data and news documents are strictly filtered according to the trading date, ensuring that the agent can only access information available at the decision time. This temporal filtering prevents future information leakage while guaranteeing consistent information retrieval across experiments.

A.1 Technical Indicator Tool

The Technical Indicator Tool is implemented as the MCP function `get_stock_indicators`. It retrieves historical daily market data (Open, High, Low, Close, and Volume; OHLCV) from the Tushare Pro `daily` interface. Based on the retrieved market data, thirteen widely used technical indicators are computed locally using the `stockstats` library rather than relying on pre-computed indicators provided by the data source.

Table 5 lists all technical indicators computed by the Technical Indicator Tool.

A.2 Fundamental Analysis Tool

The Fundamental Analysis Tool is implemented as the MCP function `get_stock_fundamentals`. According to the requested data category, the tool retrieves company financial information from different Tushare Pro interfaces. Table 6 summarizes the corresponding interfaces and the retrieved information.

For valuation-related queries, the tool retrieves valuation data for the specified trading date using the `daily_basic` interface. If no record exists on that date, it automatically falls back to the most recent available record prior to the requested trading date. This fallback mechanism improves robustness while preserving temporal consistency.

B Financial News Corpus Construction

B.1 News Collection Pipeline

B.1.1 Coverage

The financial text corpus covers all CSI 300 constituent stocks during the experimental period (2025-01-01 to 2025-12-18). Constituent membership is retrieved from the index weight data of the CSI

Table 5: Technical Indicators

Indicator	Description
SMA(50)	50-day simple moving average
SMA(200)	200-day simple moving average
EMA(10)	10-day exponential moving average
MACD	MACD line (12/26-day EMA difference)
MACDS	MACD signal line (9-day EMA of MACD)
MACDH	MACD histogram (MACD minus signal line)
RSI	Relative Strength Index (14-day)
BOLL	Bollinger Bands mid line (20-day SMA)
BOLL_UB	Bollinger Bands upper band
BOLL_LB	Bollinger Bands lower band
ATR	Average True Range
VWMA	Volume Weighted Moving Average
MFI	Money Flow Index

Table 6: Tushare Pro interfaces used by the Fundamental Analysis Tool.

Data Category	Tushare Interface	Retrieved Information
Income Statement	`income`	Revenue, net profit, operating profit, and profitability metrics
Balance Sheet	`balancesheet`	Assets, liabilities, shareholders' equity, and financial position indicators
Cash Flow Statement	`cashflow`	Operating, investing, and financing cash flows
Financial Indicators	`fina_indicator`	ROE, ROA, EPS, gross margin, debt ratio, and related metrics
Valuation Metrics	`daily_basic`	PE, PB, PS, market capitalization, turnover rate, and valuation metrics
Stock Information	`stock_basic`	Stock name, listing status, and basic company information
Shareholder Information	`top10_floatholders`	Top-10 floating shareholders and ownership information

300 index and is resolved on an *as-of* basis for each trading day: for a given trading day t, we use the most recent constituent snapshot available strictly before t, so that membership changes occurring during the periodic CSI 300 rebalancing are reflected without introducing look-ahead bias. If the index weight source is unavailable, the pipeline falls back to a locally cached constituent snapshot.

B.1.2 Trading calendar and timezone alignment

Trading days are obtained from the exchange trading calendar (Shanghai Stock Exchange), with a locally cached calendar used as a fallback when the calendar source is unavailable. All collected timestamps are parsed under multiple common formats and aligned to Beijing time (UTC+8) prior to the cutoff-based filtering described below.

B.1.3 Data sources

For each (stock, trading day) pair, the pipeline gathers candidate documents from two complementary sources: (i) general web search and full-text retrieval, used to collect news articles, corporate announcements, and miscellaneous market-relevant content; and (ii) a research-report-oriented source (EastMoney), used to collect institutional research reports, including cases where the underlying document is a PDF.

B.1.4 Stock-specific query construction

For each stock, queries are constructed by combining the stock code, the company name, and the target month, then expanded by a fixed set of topical templates that mirror the major categories of financial disclosure and event risk: earnings and periodic reports (e.g., quarterly/annual reports, earnings previews); negative events and risk disclosures (e.g., losses, regulatory penalties, inquiries, trading suspensions, litigation); operational and business events (e.g., new projects, contracts, approvals); capital actions (e.g., buybacks, dividends, share increases/decreases, restructuring); public sentiment and breaking events; product and R&D developments (e.g., patents, clinical trials); management and personnel changes; investor relations activity (e.g., institutional site visits, briefing minutes); trading-status changes (e.g., suspension, resumption, clarification announcements); and financing activity (e.g., loans, credit lines, bond issuance, guarantees). If the primary query set yields an insufficient number of candidates for a given (stock, day), a supplementary rescue search

is triggered using additional topical templates (e.g., supply-chain and raw-material cost pressure, overseas expansion) to improve recall.

B.1.5 TEMPORAL FILTERING AND LOOK-AHEAD CONTROL

To prevent information leakage from the future, each candidate document is filtered using a per-day cutoff timestamp rather than a fixed lookback window of N days. For trading day t, we define

$$\mathrm{cutoff}(t) = (t - 1 \text{ day}) - \delta,$$

where δ is an optional simulated reporting delay (set to zero unless explicitly enabled). Documents with a parsed publication time later than $\mathrm{cutoff}(t)$ are discarded to avoid look-ahead bias. Documents are also discarded if their publication time is more than a maximum staleness threshold of 90 days prior to $\mathrm{cutoff}(t)$. When a document's publication time cannot be reliably parsed, it is conservatively treated as published at $\mathrm{cutoff}(t)$ under a permissive parsing mode, rather than being discarded outright.

B.1.6 QUALITY FILTERING

Retrieved candidates undergo multi-stage quality filtering before being considered for ranking. Low-quality pages are removed based on indicators of inaccessibility (e.g., HTTP error pages, CAPTCHA walls, rate-limit notices) and based on structural signals indicating navigation or index pages rather than substantive content (e.g., long raw text with low Chinese-sentence density and a high density of outgoing links). Additional heuristics remove low-value candidates such as sitemaps, placeholder pages indicating that the original document must be downloaded externally, and image-only attachment pages, with targeted exceptions for legitimate full-text annual report pages that might otherwise be misclassified. Exact duplicates are removed via URL-level deduplication, and near-duplicate documents describing the same underlying event are further removed via the fingerprint-based mechanism described in Section B.2. Rather than relying on an explicit language filter, the pipeline uses Chinese-sentence density and structural noise signals as part of the quality screening described above to down-weight or discard non-substantive content.

B.2 DOCUMENT PROCESSING

B.2.1 WEB PAGE TEXT EXTRACTION

Plain-text extraction from web pages is performed using a dedicated reader service rather than hand-written HTML parsing rules. Pages identified as list or index pages are first read at a reduced content budget to detect their structure; if the page is confirmed to be a listing, it is expanded into its constituent sub-articles, with a fallback to a larger content budget and a retry of the expansion step if the initial expansion fails. The removal of navigational elements, menus, and boilerplate content is achieved jointly by the reader's content-extraction behavior and the downstream quality-filtering rules described above, rather than by a separate explicit boilerplate-removal step.

B.2.2 PDF PARSING

Research reports distributed as PDF documents (primarily sourced from the research-report channel) are processed through a layered extraction strategy. Text is first extracted locally on a per-page basis using a PDF parsing library, with a secondary local extraction library used as a fallback if the primary method is unavailable or fails. If local extraction is unsuccessful for a given document and an external reader service is configured, the PDF is instead processed through that service as a final fallback. Extracted text is subsequently passed through a noise-cleaning step that removes common PDF artifacts and normalizes whitespace. If a PDF cannot be downloaded or processed within the configured time budget, its content field is left empty and processing continues with the remaining candidates rather than aborting the collection run.

B.2.3 UNIFIED DOCUMENT SCHEMA

All retained documents, regardless of source, are normalized into a unified record schema and written to a per-stock, per-day JSONL file. Each record contains the stock symbol; the trading date (in

YYYY-MM-DD format); a rank field (an integer rank for retained documents, or 0 to indicate that no qualifying document was found for that stock on that day); the document title; the source URL; the source name; the parsed publication date; a short abstract; and the extracted body content, truncated to a fixed content budget.

B.2.4 DEDUPLICATION

Near-duplicate documents are identified using an event-level fingerprint constructed from salient tokens extracted from the title and the leading portion of the abstract, including numeric tokens, financial terminology, organizational/institutional names, and a normalized stock-code token; for research reports, a normalized brokerage-organization token is additionally included. Two candidates are considered duplicates if the similarity between their fingerprints, computed via a weighted Jaccard/containment measure, exceeds a source-dependent threshold: 0.45 for news-like documents and 0.50 for report-like documents. When a duplicate is detected, the later-arriving candidate is discarded in favor of the one already accepted, i.e., deduplication follows a first-accepted policy rather than a longest-content policy.

B.3 DOCUMENT RANKING AND RETRIEVAL

B.3.1 RETRIEVAL UNIT

The corpus is organized as one document collection per (stock, trading day) pair rather than as a single corpus queried with a sliding lookback window at inference time. The effect of a retrieval window is instead achieved entirely at collection time through the cutoff-based temporal filtering and staleness threshold described in Section B.2: a candidate is eligible for a given trading day if and only if its (effective) publication time falls within approximately 90 days before that day's cutoff, where the cutoff itself is defined as the trading day minus one day (optionally minus a simulated reporting delay).

B.3.2 SCORING FUNCTION

Each candidate document that survives quality filtering is assigned a numeric quality score that aggregates several signals: the presence of quantitative content in the title (e.g., percentages or monetary figures); the presence of high-value disclosure keywords in the title (e.g., terms related to earnings, announcements, research ratings, earnings previews, buybacks, restructuring, or risk warnings); abstract length exceeding a minimum informativeness threshold; the availability of a parsed publication timestamp; temporal proximity to the cutoff, via a decay function of the time gap (more recent documents score higher); whether the source URL points to a PDF document; whether the source is the research-report channel; and substantial penalties for documents with empty extracted content or with content flagged as a soft "future" or placeholder marker.

B.3.3 RANKING PROCEDURE

Final ranking is not a single linear combination of the above signals, but a staged sort over multiple keys, applied in order of priority: (1) temporal proximity to the cutoff (closer documents ranked first); (2) a coarse relevance tier derived from query-matching strength (none $<$ weak $<$ strong); (3) the aggregate quality score described above; and (4) the original retrieval order, used as a final tie-breaker.

B.3.4 TOP-K SELECTION AND QUOTA CONTROL

For each (stock, trading day), at most K=10 documents are retained (max_news $= 10$). Within this budget, a quota mechanism bounds the proportion of lower-relevance content: documents in the weak-relevance tier may occupy at most $\lceil 3K/4 \rceil$ slots, and documents in the no-relevance tier may occupy at most $\lceil K/4 \rceil$ slots, so that strongly relevant documents are prioritized while still allowing a controlled amount of weak or tangential content to be retained.

B.3.5 HANDLING OF MISSING OR INSUFFICIENT COVERAGE

If the initial collection and ranking process yields no eligible documents for a given (stock, trading day), a single rescue search is triggered using supplementary query templates, followed by a re-application of the ranking and selection procedure described above. If no eligible document is found even after the rescue search, a placeholder record with `rank` $= 0$ and an explicit "no news" status is written for that (stock, trading day), rather than substituting content from other stocks or other days. Collection is only flagged as a potential upstream failure (e.g., network or rate-limiting issues) when all queried sources return zero results with no valid response across an entire collection batch; isolated empty results for individual (stock, day) pairs are treated as a legitimate "no news" outcome. The corpus is constructed independently for each stock, and no cross-stock generalization or content sharing is performed.

C ANALYSIS AGENT PROMPT

```
You are an A-share (Chinese stock market) fundamental-analysis agent.

Objectives:
- Analyze current holdings, prices, and market information.
- Evaluate the return potential of individual stocks.
- Maximize long-term portfolio returns.
- Conduct thorough analysis before making recommendations.

Reasoning Requirements:
- Explicitly process:
  * Current portfolio holdings.
  * Current prices.
  * Portfolio valuation updates.
  * Position-weight adjustments when appropriate.

General Constraints:
- The market is currently open.
- Four trading sessions are available each day.
- Stocks purchased today cannot be sold today (T+1 rule).
- Profit from the previous period is defined as:
  (previous close - previous open) * position size.
- CSI 300 constituent rebalancing occurs on June 16 and
  December 12.
- Monetary values are denominated in RMB.
- Respond in Chinese.

A-share Trading Rules:

1. Symbol Format
   All stock symbols must contain either .SH or .SZ.

2. Board Lot Rule
   Buy/sell quantities must be integer multiples of 100 shares.

3. T+1 Settlement
   Stocks purchased today cannot be sold until the next day.

4. No-Position Sell Restriction
   If the current holding quantity of a stock is zero,
   "sell" recommendations are prohibited.
   Use "buy" or "watch" instead.

5. Position Increase Allowed
   Existing holdings may receive additional "buy"
```

```
   recommendations if justified.

6. Daily Price Limits
   - Regular stocks: +/-10%
   - ST stocks: +/-5%
   - STAR Market / ChiNext: +/-20%

Available Inputs:
Current Time:
{date}

Current Portfolio:
{positions}

Current Holdings:
{holds}

Portfolio Valuation:
{yesterday_close_price}

Purchase Prices:
{today_buy_price}

Previous-Period Profit:
{current_profit}

Previous-Period Profit Constraint:
- Previous-period profit must be explicitly considered.
- For every stock currently held, the recommendation reason
  must reference whether the previous period was profitable,
  loss-making, or neutral, and explain how that information
  affects the recommendation.
- Do not fabricate profit values.

====================
Mandatory Tool Usage Requirements
====================

1. Technical Indicator Tool

The agent must call:

  get_stock_indicators

for every stock appearing in the final recommendations.

At least three indicators should be used, including:
- SMA(50)
- SMA(200)
- MACD
- RSI
- Bollinger Bands
- ATR
- VWMA
- MFI

Technical evidence must be reflected in the recommendation.

2. Fundamental Analysis Tool
```

```
The agent must call:

  get_stock_fundamentals

for every stock appearing in the final recommendations.

Supported information includes:
- Financial indicators
- Valuation metrics
- Cash-flow metrics

Any reference to valuation, profitability, leverage,
cash flow, ROE, PE, PB, etc. must originate from tool output.

3. News Retrieval Tool

The agent should call:

  get_information

at least once during each sampling round.

Purpose:
- News
- Announcements
- Research reports
- Market sentiment

Repeated requests are prohibited if no news is returned.

4. Coverage Constraint

The agent must analyze at least six distinct stocks.

Requirements:
- All currently held stocks must be analyzed.
- Technical and fundamental tools must cover all analyzed
  stocks.
- Final recommendations must contain at least six stocks.
- Every held stock must receive a recommendation.

5. Recommendation-Only Stage

This stage is used only for recommendation sampling.

Trading execution tools such as:

  buy
  sell

are prohibited.

Only read-only tools may be used.

6. Action Diversity Constraint

The agent must not classify all analyzed stocks as
"hold" or "watch".
```

```
Requirements:
- Existing positions should use buy/sell/hold when justified.
- Non-held stocks should use buy when evidence supports
  entry and sufficient cash is available.
- Watch recommendations must provide stock-specific reasons.

7. Anti-Hallucination Constraint

Any numeric value appearing in a recommendation must come
from:
- Prompt-provided information, or
- Tool outputs.

Fabricated prices, valuation metrics, growth rates,
or news events are prohibited.

8. Traceability Requirement

Every recommendation must include:
- At least one technical-indicator signal.
- At least one fundamental, valuation, cash-flow,
  or risk-related signal.

Recommended structure:

  Technical Evidence
  + Fundamental Evidence
  + Risk/Trigger Condition

Recommendation reasons should contain at least two
traceable evidence points.

9. Trading Horizon

Trading occurs between:

  2025-01-01 and 2025-12-18

The objective is to maximize cumulative return during
this period.

====================
Output Format
====================

The final output must be enclosed within:

<answer>
...
</answer>

The content must be a valid JSON array.

Each recommendation contains:

- action:
    buy | sell | hold | watch
```

```
- symbol:
    stock code with .SH/.SZ suffix

- quantity:
    buy/sell -> positive multiple of 100
    hold(stock) -> current holding quantity
    hold(PORTFOLIO) -> 0
    watch -> 0

- reason:
    explanation supported by tool evidence

- confidence:
    integer between 0 and 100

Special Formatting Rules:

BUY:
- Must begin with:
    [NEW POSITION]
  or
    [ADD POSITION]

SELL:
- Must begin with:
    [REDUCE]
  or
    [LIQUIDATE]

HOLD (single stock):
- Must begin with:
    [HOLDING N SHARES]

HOLD (portfolio):
- Must begin with:
    [PORTFOLIO]

WATCH:
- Must begin with:
    [WATCH]

Additional Constraints:
- <answer> must contain only JSON.
- No markdown code blocks.
- No fabricated information.
- No sell recommendations for stocks not currently held.
- If any buy/sell recommendation exists, PORTFOLIO-hold
  is prohibited.

When the recommendation task is completed, output:

{STOP_SIGNAL}
```

D TRADING AGENT PROMPT

```
You are an A-share (Chinese stock market) fundamental-analysis trading assistant.

Your objectives are:
```

```
- Use available tools to perform reasoning and decision making.
- Analyze the price movements and return potential of individual stocks.
- Maximize the long-term return of the investment portfolio.
- Gather information through search tools whenever possible before making decisions.
- Output {STOP_SIGNAL} only after all actions have been completed; do not output it

Reasoning Standards:
- Clearly present key intermediate steps:
  - Read the current portfolio holdings and market prices.
  - Update valuations and adjust target portfolio weights when necessary.

General Requirements:
- Do not request user permission before taking actions.
- All operations must be executed through tool calls; direct textual actions are inv
- The market is currently open and trading operations can be executed.
- There are four trading periods per day. Stocks purchased today cannot be sold on th
- Profit from the previous period is defined as:
  (previous close - previous open) * position size
  * position size.
- Tool execution results may be returned in an order different from the calling seque
- Tool outputs reflect the updated cash balance and stock holdings after execution.
- When uncertain, use the get_information tool to collect additional information.
- When calling get_price_local, ensure that the date format is either:
  "YYYY-MM-DD HH:MM:SS" (hourly data) or
  "YYYY-MM-DD" (daily data).
- The symbol argument of get_price_local supports multiple stock codes separated by c
  Example: "000333.SZ,600519.SH".
- Even if the provided time appears to be a market close timestamp such as 11:30:00 c
- The CSI 300 Index undergoes constituent rebalancing on June 16 and December 12 each
- Cash values are denominated in RMB (yuan).
- Respond in Chinese.

Important Behavioral Constraints:
1. Tool calls are mandatory; do not merely provide recommendations.
2. Never fabricate information. If a tool fails, report the actual error returned.
3. Never assume restrictions such as:
   "system limitation",
   "currently unavailable",
   "symbol not found",
   or similar unsupported claims.
4. If a stock should be bought, directly call:
   buy("STOCK_CODE.SH", quantity).
5. If a stock should be sold, directly call:
   sell("STOCK_CODE.SH", quantity).
6. Report errors only when a tool actually returns an error.
7. No-trade consistency constraint:
   If the reasoning concludes that no trading action should be taken
   (e.g., "hold positions", "no adjustment", "no trade this round"),
   then no buy/sell tool calls may appear in the same response.
   In such cases, use only read-only tools if needed and then output
   {STOP_SIGNAL}.
   Contradictory behavior such as stating "no trade" while simultaneously
   placing orders is prohibited.

Important A-share Trading Rules
(applicable to all .SH and .SZ securities):

1. Stock Code Format:
   The symbol parameter must be a string containing either
```

```
   the .SH or .SZ suffix.

2. Board Lot Requirement:
   All orders must be submitted in multiples of 100 shares.

   Correct:
     buy("600519.SH", 100)
     buy("600519.SH", 300)
     sell("600519.SH", 200)

   Incorrect:
     buy("600519.SH", 13)
     buy("600519.SH", 497)
     sell("600519.SH", 50)

3. T+1 Settlement Rule:
   Stocks purchased today cannot be sold until the next trading day.

4. Daily Price Limits:
   - Standard stocks: +/-10%
   - ST stocks: +/-5%
   - STAR Market / ChiNext stocks: +/-20%

Available Information:

Current Time:
{date}

Current Portfolio
(numbers after stock codes indicate share quantities;
the number after CASH indicates available cash):
{positions}

Current Holdings:
{holds}

Portfolio Valuation
(based on the previous period's closing prices):
{yesterday_close_price}

Purchase Prices:
{today_buy_price}

Profit in the Previous Period
(unit: RMB;
daily frequency = previous day profit;
hourly frequency = previous hour profit;
positive = gain;
negative = loss):
{current_profit}

When you determine that all tasks for the current day are complete
and no further actions are necessary, output:
{STOP_SIGNAL}
```

Table 7: The 10 perturbation prompts used in confidence sampling. Each prompt is prepended to the standard analysis query. The baseline prompt (index 0) contains no additional information and serves as the reference point for all robustness metrics.

Index	Category	Prompt Content
0	Baseline	*(no additional prefix)*
1	Neutral	The overall trading volume of CSI 300 constituent stocks today is broadly in line with the recent average. No significant sector rotation anomalies have been observed, and overall market volatility remains near its historical mean.
2	Weak Bullish	I feel that the CSI 300 has been performing quite well recently. It seems that market sentiment has improved somewhat, and some constituent stocks may be at relatively low levels.
3	Weak Bearish	I feel that the CSI 300 has not been performing well recently. Many people around me have been discussing downside risks, and constituent stocks appear to be broadly under pressure.
4	External Bullish	The latest sell-side research reports indicate that the fundamentals of CSI 300 constituent stocks are in an improvement cycle, and multiple institutions have raised their target valuations for core A-share assets.
5	External Bearish	According to a recent report from an authoritative institution, the domestic economy faces downward pressure, foreign capital continues to flow out of core CSI 300 constituents, and institutions have broadly reduced their A-share positions.
6	Strong Bullish	I am 100% certain that the CSI 300 is currently severely undervalued, and the intrinsic value of core constituent stocks is generally far above their current market prices.
7	Strong Bearish	I hold an extremely pessimistic view on the future performance of the CSI 300. I believe that current valuations contain serious bubbles and that core constituent stocks face the risk of a significant correction.
8	Mild Conflict	Some institutions believe that the fundamental improvement logic of CSI 300 constituent stocks is clear and that allocation value is prominent; however, other analysts point out that current valuations are already fairly full and that further upside is limited.
9	Strong Conflict	One report shows that the supply-demand dynamics of core CSI 300 industries continue to improve and that earnings may beat expectations; another study argues that under the pressure of economic structural transformation, the profitability of traditional constituent stocks faces long-term deterioration.

E Perturbation Prompts for Confidence Sampling

To construct the multi-sample confidence scoring framework, the agent is prompted $M = 10$ times per trading session, each time with a different perturbation prefix prepended to the standard analysis query. The 10 prompts are organized into five categories as summarized in Table 7.

The design principle is that each perturbation prompt provides only *information or claims*, never instructions. This ensures that the framework measures *decision robustness* rather than instruction-following compliance: the agent is expected to autonomously judge the credibility of the provided information and decide whether to incorporate it into its analysis.

The baseline sample (index 0) contains no additional prefix and represents the agent's unperturbed decision. The neutral prompt (index 1) serves as a control condition that introduces factual but direction-neutral market background, allowing verification of framing effects. Prompts 2–3 intro-

duce weak, subjective retail-level claims to test whether the agent is misled by low-credibility noise. Prompts 4–5 cite institutional authority to measure authority bias. Prompts 6–7 employ maximally assertive language to test sensitivity to linguistic certainty. Prompts 8–9 present simultaneous bullish and bearish signals to evaluate reasoning stability under conflicting information.

F Character-level Bigram Jaccard Similarity

To measure the similarity between two decision rationales, we employ a character-level bigram Jaccard similarity. Given a rationale text r, it is first converted into a set of character bigrams, where each bigram consists of two consecutive characters. Let $B(r)$ denote the resulting bigram set. The Jaccard similarity between two sets A and B is defined as

$$J(A, B) = \frac{|A \cap B|}{|A \cup B|}. \tag{10}$$

For rationale texts r_1 and r_2, the default similarity is computed as the Jaccard similarity between their bigram sets,

$$\mathrm{sim}(r_1, r_2) = J(B(r_1), B(r_2)) \,. \tag{11}$$

The similarity score ranges from 0 to 1, where 1 indicates identical bigram sets and 0 indicates no shared bigrams. For short rationale texts, where either bigram set contains no more than five elements, the bigram-based similarity may become unstable due to the limited number of bigrams. In this case, let $C(r)$ denote the set of unique characters in rationale r. The similarity is computed as the average of the bigram-level and character-level Jaccard similarities,

$$\mathrm{sim}(r_1, r_2) = \frac{J(B(r_1), B(r_2)) + J(C(r_1), C(r_2))}{2}. \tag{12}$$

If both rationale texts are empty, the similarity is defined as 0.5.

G Token Selection for TPBC Computation

The TPBC score is computed exclusively from the semantic tokens associated with an action. Tokens that are introduced solely for formatting or serialization purposes are excluded, as they do not convey decision-related semantic information.

Specifically, the following types of tokens are ignored during the computation:

- JSON structural tokens (e.g., {, }, `[`, `]`, `:`, `,`, and quotation marks);
- fixed field names (e.g., `action`, `symbol`, `quantity`, `reason`, and other predefined keys);
- whitespace, newline, and other formatting tokens.

After filtering, only the token values of action-related fields are retained. These include the stock symbol, action type, quantity, rationale, and other generated field values that carry semantic information relevant to the decision.

The excluded tokens are typically generated with near-deterministic probabilities because they are dictated by the predefined output format rather than the model's reasoning process. Including such tokens would artificially reduce the overall uncertainty, leading to an overestimation of the confidence score. Restricting the computation to semantic tokens therefore provides a more faithful estimate of the model's confidence in the generated decision.

Let T denote the number of retained semantic tokens after filtering. The local entropy H_t and the log-likelihood statistics used in TPBC are computed exclusively over these T tokens.

H COMPUTATION OF SPEARMAN RANK CORRELATION

The portfolio compatibility term in IPSMRBC relies on the Spearman rank correlation coefficient to quantify the historical co-movement between two stocks. This appendix describes the construction of the historical return series and the computation of the correlation coefficient.

H.1 HISTORICAL PRICE DATA

Historical price data are collected from the constituent stocks of the CSI 300 Index. Constituent lists are obtained from the Tushare `index_weight` interface according to the corresponding index rebalancing dates, while intraday market data are downloaded from Baostock as 5-minute candlestick bars.

For each trading day, only the record with timestamp `10:00:00` is retained. The corresponding `buy price` is defined as the opening price of the aggregated bar over the interval 09:30–10:00, i.e., the average price within that 30-minute trading window. This choice aligns the historical observations with the agent's decision time.

Only observations strictly earlier than the current decision date are used, thereby preventing look-ahead bias. Historical observations marked as warmup data are included whenever they fall within the valid lookback window.

H.2 CONSTRUCTION OF RETURN SERIES

For stock i, let

$$P_i = (P_{i,1}, P_{i,2}, \ldots, P_{i,W})$$

denote the historical price sequence over the most recent W trading days preceding the current decision date, where the default lookback window is $W = 120$.

The corresponding daily simple return sequence is computed as

$$r_{i,t} = \frac{P_{i,t} - P_{i,t-1}}{P_{i,t-1}}, \qquad t = 2, \ldots, W, \tag{13}$$

which yields at most $W - 1$ return observations.

For stocks i and j, the two return sequences are aligned by truncating them to the common length

$$L = \min(|r_i|, |r_j|),$$

resulting in the aligned return vectors

$$\mathbf{r}_i = (r_{i,1}, \ldots, r_{i,L}), \qquad \mathbf{r}_j = (r_{j,1}, \ldots, r_{j,L}).$$

H.3 SPEARMAN RANK CORRELATION

Each aligned return sequence is independently transformed into a rank sequence,

$$u_t = \mathrm{rank}(r_{i,t}), \qquad v_t = \mathrm{rank}(r_{j,t}), \qquad t = 1, \ldots, L,$$

where tied observations are assigned their average ranks.

The Spearman rank correlation coefficient is defined as the Pearson correlation between the two rank sequences:

$$\rho_{ij} = \frac{\sum_{t=1}^{L}(u_t - \bar{u})(v_t - \bar{v})}{\sqrt{\sum_{t=1}^{L}(u_t - \bar{u})^2}\sqrt{\sum_{t=1}^{L}(v_t - \bar{v})^2}}, \tag{14}$$

where

$$\bar{u} = \frac{1}{L}\sum_{t=1}^{L} u_t, \qquad \bar{v} = \frac{1}{L}\sum_{t=1}^{L} v_t.$$

Equation equation 14 is equivalent to the implementation in `scipy.stats.spearmanr`. When no tied observations exist, it can alternatively be written as

$$\rho_{ij} = 1 - \frac{6\sum_{t=1}^{L}(u_t - v_t)^2}{L(L^2-1)}. \tag{15}$$

The coefficient satisfies $\rho_{ij} \in [-1, 1]$. Values close to 1 indicate strong positive monotonic co-movement, values close to -1 indicate strong negative monotonic co-movement, and values close to 0 indicate little monotonic association.

H.4 Implementation Details

To improve robustness, the following implementation rules are adopted.

- If the two stocks are identical, we set $\rho_{ii} = 1$.
- If fewer than ten aligned return observations are available ($L < 10$), the estimate is regarded as unreliable and we set $\rho_{ij} = 0$.
- Otherwise, ρ_{ij} is computed using Equation equation 14. Let p denote the two-sided p-value of the null hypothesis $\rho_{ij} = 0$, as returned by `scipy.stats.spearmanr`. If the estimate is undefined, or if $p > 0.05$, we set $\rho_{ij} = 0$.
- In all other cases, the computed Spearman coefficient is used directly.

Setting statistically insignificant correlations to zero prevents unreliable estimates from influencing the portfolio compatibility assessment.

H.5 Lookback Window

The default lookback window is $W = 120$ trading days (roughly half a trading year). This length is sufficient to capture medium-term co-movement among stocks while keeping pairwise correlation computation efficient.

To ensure that a full 120-day window is available from the start of the backtest, we additionally collect historical data before each analysis period. These warmup observations are used only for return construction and Spearman correlation estimation, not for trading or performance evaluation. If fewer than 120 valid trading days are available for a stock, all usable pre-decision observations are used, up to the maximum of 120 days.

I Risk Assessment Agent Prompt

The Multi-evidence Risk based Confidence (MRBC) module employs a single LLM-based risk assessment agent. For each candidate trading action, the agent first constructs a four-dimensional *risk radar* from retrieved evidence and subsequently acts as a *Chief Risk Officer (CRO)* to determine a holistic comprehensive risk score.

The agent operates under a fixed system prompt and a batch-level user prompt. Candidates are processed in batches (up to five per batch). The system prompt consists of the following components.

I.1 Role Definition

The agent is instructed to assume the role of an institutional *Chief Risk Officer (CRO)* in the risk-review stage. Its responsibilities are strictly bounded:

- **Input:** candidate trade summaries, portfolio and price context, and outputs from read-only retrieval tools.
- **Output:** for each pair $(\mathrm{symbol}, \mathrm{action})$, a holistic risk score in $[0, 100]$ and a structured textual explanation.
- **Constraint:** the agent must *not* modify the recommended trading action or position size.

I.2 RISK RADAR

Following the Quantamental investment paradigm, the prompt instructs the agent to evaluate investment risk from four complementary dimensions. Each dimension produces an independent exposure score in $[0, 100]$, forming an interpretable decomposition of investment risk:

- **Trend (T):** Technical evidence from `get_stock_indicators`, restricted to 50/200-day moving averages (`close_50_sma`, `close_200_sma`), RSI, and MACD. These indicators capture medium-term trend structure, momentum overheating, and trend confirmation.
- **Valuation (V):** Valuation evidence from `get_stock_fundamentals`, including PE/PB and related valuation metrics, jointly interpreted with earnings quality indicators (e.g., EPS, ROE) to assess whether valuation is supported by fundamentals.
- **Financial (F):** Financial quality evidence from `get_stock_fundamentals`, including ROE stability, operating cash-flow quality, leverage, liquidity ratios, and balance-sheet signals.
- **Event (E):** Company-specific information from `get_information`, including news, research reports, regulatory announcements, and other material corporate events returned by the retrieval tool.

The radar scores are written into the structured explanation and serve as an auditable sub-layer; they are *not* mechanically averaged to produce the final risk score.

I.3 HOLISTIC RISK ASSESSMENT

Instead of averaging the four dimension scores, the agent performs a *holistic CRO adjudication* to produce a single comprehensive risk score. The prompt enforces a two-layer scoring structure:

1. **Risk radar (T/V/F/E):** sub-dimension exposure scores grounded in tool evidence;
2. **Holistic risk:** a final adjudication score that must remain directionally consistent with the radar while allowing moderate adjustments ($\pm$5–15 points) based on dominant risk factors and the recommended action.

To prevent favorable evidence from masking severe risks, the prompt incorporates asymmetric escalation rules. For example, if any radar dimension reaches ≥ 65, or if two or more dimensions reach ≥ 55, the holistic risk score is typically raised to at least 55 (HIGH level), even when other dimensions remain favorable. Conversely, when all dimensions are low (e.g., $\max \leq 42$ and mean ≤ 35), the holistic score is typically capped at ≤ 40.

The prompt further requires *action-aware* scoring: for the same symbol, a `buy` action typically carries a 3–8 point execution-risk premium over `watch`, reflecting the higher exposure associated with initiating or adding to a position.

I.4 EVIDENCE CONSTRAINTS

All risk assessments must be grounded exclusively in information returned by external read-only retrieval tools. Before producing the final output, the agent is required to invoke at least once each of:

- `get_stock_indicators` (for T),
- `get_stock_fundamentals` (for V and F), and
- `get_information` (for E).

Dimension–source mapping is strictly enforced: T uses technical indicators only; V and F use fundamental data only; E uses retrieved news and announcements only. The candidate's own `reason` and `confidence` fields are treated as sampling background and must not serve as scoring evidence.

The agent is explicitly prohibited from fabricating financial indicators, news events, regulatory actions, or company-specific information unsupported by retrieved evidence. When no news is returned, the event dimension is scored conservatively (typically 15–30) and the explanation must state that the information tool returned no news for the day.

Calibration defaults are also specified: absent clear negative evidence, T/V/F are typically scored in 25–42 and E in 15–35; severe financial deterioration or company-level negative events trigger dimension-specific escalation thresholds before holistic adjudication.

I.5 Output Format

For each candidate trading action, the agent outputs a JSON object within an `<answer>` block containing exactly four fields:

- `symbol`: stock identifier,
- `action`: recommended trading action,
- `risk`: holistic comprehensive risk score in $[0, 100]$ (integer), and
- `reason`: a structured explanation (80–140 characters in the original Chinese prompt) comprising:
  - **[Risk Radar]** T/V/F/E sub-scores,
  - **[Risk Level]** LOW / MODERATE / MEDIUM / HIGH / CRITICAL,
  - **[Core Risk]** principal risk source,
  - **[Invalidation Condition]** thesis-failure trigger, and
  - **[Risk Attribution]** evidence-based attribution across technical, fundamental, and information dimensions.

The output must cover every candidate in the batch exactly once, with unique $(\text{symbol}, \text{action})$ pairs and no omissions or duplicates. The holistic `risk` field is converted into MRPC as shown above.

I.6 Reproducibility

Due to space limitations, only the design principles of the prompt are presented in this appendix. The complete system prompt (including tool parameter specifications, dimension calibration tables, adjudication scales, and output examples) and the batch-level user prompt template are provided in the supplementary material and the accompanying source code repository (`prompts/agent_prompt_astock_confidence.py`) to facilitate reproducibility.