ingegneria sismica
International Journal of Earthquake Engineering
Anno XLIII - Num. 3 - 2026

FinGuard: An Adaptive Knowledge-Stratified Multi-Agent System for Stability and Compliance in Stock Trading

Chengyi Peng[1,*]

[1] School of Physics and Electronic Information Engineering, Henan Polytechnic University, Jiaozuo 454003, China

SUMMARY: *High-risk Stock trading needs to meet the above criteria for strict compliance and stable long-term returns. However, the existing methods generally combine market information and compliance rules in a single probabilistic retrieval-and-generation process, thereby weakening the constraint effect of compliance during long-chain reasoning and causing cascading error amplification. FinGuard is a multi-agent system that can reduce the number of high-risk stocks through the use of knowledge stratification. First, the continuously learning discriminator dynamically identifies and directs knowledge of different natures. Constraint-as-Code transforms rigid rules into executable logic, and can thus perform more explicit and consistent rule evaluation in the current system. A sliding safety audit mechanism continuously observes intermediate states in long-chain reasoning to reduce error accumulation and error propagation. In a controlled stock-trading evaluation over 1,764 trading days, FinGuard has shown stable results at the levels of day, week and month. FinGuard is better than the ReAct baseline and has increased compliance by 16.7% and adjusted returns by 14.7%. Stress tests show that both the drop and fluctuation are significantly smaller; therefore, FinGuard will meet the conditions for strict compliance and stable finances.*

KEYWORDS: *high-risk stock trading; Agentic RAG; multi-agent systems; compliance constraints; adaptive knowledge stratification; risk-adjusted return*

1 Introduction

As artificial intelligence has been applied to finance and other areas in recent years, so too has the problem of making high-risk decisions under uncertainty[1-3]. Financial decision-making needs to achieve both stable long-term returns and full compliance with regulatory requirements at the same time; otherwise, it would be classified as a general knowledge-based task [4]. Such tasks need to handle changes in market conditions, updates to information at various times, and decision-making under the risk of significant losses from a single error [5]. Therefore, this setting places higher demands on the continuous reasoning and risk-control capabilities of a model. Unlike general knowledge-based question answering [6], stock trading frequently needs to obtain new information, adjust one's assessment accordingly, and take corresponding actions through multiple rounds of communication and repeatedly performing these actions [7]. Even a small error in the initial reason will gradually accumulate in all the following steps, thus affecting both the return rate and the compliance risk [8]. The objectives of the compliance rules for return optimisation are not flexible; they must be realised

---

*17681048969@163.com

https://doi.org/10.65102/is20261267

consistently across the board.

Agentic RAG [9] is a typical decision-making framework for large language models that combines retrieval augmentation and multi-step planning to provide a new path for automating stock trading [10]. By integrating external knowledge in the generation stage [11], the above methods can utilise market news, stock prices and other data for advanced reasoning and expand the scope of decision-making. The main problem is that most of the existing systems still process dynamic market signals and static compliance rules in the same retrieval-generation loop [12], implicitly assuming that both can be retrieved, interpreted, and applied in the same way. In a high-risk financial situation, this will reduce the importance of rule-based constraints and lead to a dilution of compliance information amidst market noise in long chains of reasoning [13]. Therefore, the system cannot ensure that the decisions are uniform and provide proof of compliance for all output steps.

The three main directions of the existing research on this issue have developed along the following paths. The first type of method enhances retrieval consistency by using knowledge graphs or other structured representations, thus reducing factual bias in the application of external knowledge; the second group is based on self-correction [14], multi-agent collaboration [15-17], and cross-validation to fix local errors during generation; the third type attempts to turn some inference steps into program execution to improve the controllability of single-step decision-making [18-21]. Although these efforts have improved retrieval quality, generation robustness and local verifiability individually, most of them have focused on optimising a single aspect [22, 23] and have yet to address the two more serious problems simultaneously: how dynamic information and rigid compliance rules should be handled differently at the system level, and how early biases in long-chain decision-making should be continuously mitigated [24-27]. Therefore, the current methods have not been able to balance the requirements for long-term performance and strict compliance in high-risk financial scenarios effectively.

Based on the problems and observations mentioned above, this paper proposes FinGuard, an adaptive knowledge-stratified Agentic RAG framework for high-risk stock trading. The first concept of its design does not combine market information and regulatory requirements in a single generation process; rather, it is divided into soft knowledge and hard knowledge (3.2) at the system level and has different processing paths for each. Continuously Learning Discrimination is responsible for semantic decomposition and knowledge streaming of the input; Constraint-as-Code compiles executable compliance rules into deterministic constraint logic; and Sliding Safety Audit continuously reviews intermediate states and candidate actions during long-chain reasoning. The above design reduces the conflict between the flexibility of information and the certainty of rules; thus, a more stable foundation for long-term, reliable decisions has been established.

A total of 1,764 trading days of a controlled stock trading evaluation were analysed for the performance of the framework at three levels of granularity: daily, weekly and monthly. Based on the experimental results, FinGuard has shown better performance than the above baseline models ReAct, CoT, and Self-Refine in terms of compliance rates and adjusted returns in our environment; furthermore, after the addition of non-compliance penalties, it has demonstrated a more stable cumulative return path. Further temporal stress tests and ablation experiments have shown that the above advantages do not result from more aggressive, high-risk decisions but rather from the combined effects of knowledge routing, constraint adjudication, and process auditing. The main goal of this paper is to enhance the prediction performance of a single point in time for returns and, simultaneously, to present a feasible implementation solution for Agentic RAG systems in high-risk financial environments that meets the demands for long-term stability and stringent compliance. The first and other are listed below.

1. FinGuard: An Agentic RAG Framework for Adaptive Knowledge Stratification in High-Risk Stock Trading. FinGuard systematically differentiates and cooperatively processes knowledge of different natures, routes soft and hard knowledge through separate reasoning and execution paths, and enhances the priority of compliance constraints.
2. We have developed an adaptive knowledge-stratified agentic reasoning model to distinguish between soft and hard knowledge and set up two independent decision-making paths to ensure stable enforcement of compliance and adaptability to fluctuating information.
3. We design Constraint-as-Code and Sliding Safety Audit to convert compliance rules into executable constraint logic and continuously inspect intermediate states during long-horizon reasoning to effectively suppress error accumulation in multi-round reasoning and multi-agent collaboration.
4. Many experiments show that FinGuard significantly reduces the number and size of violations while keeping risk-adjusted returns at a high level. The reason is that CLD, CaC and SSA are not considered high-risk or non-compliant, so they can be combined.

2 Related Work on Financial Agent Systems

2.1 RAG and Structured Retrieval for Stock Trading

Large Language Models and retrieval-enhanced methods for financial applications provide support to assist in decision-making. Recent finance-oriented studies on LLMs and agents have improved the model for specialised terminology, contextual semantics and deployment constraints through adaptation to financial corpora and financial task settings [28, 29]. RAG expands the limited parameter memory by including external documents. At the same time, some research has begun to introduce knowledge graphs and other structured representations in the retrieval process to enhance the factual accuracy and interpretability of retrieval results via entity relationships. Although this approach enhances the accuracy of external knowledge integration, it mainly aims to improve information retrieval and the quality of evidence; it has not been divided according to different functions in the decision-making process [30]. In high-risk financial environments, the above methods generally process dynamic market signals and rigid compliance regulations in a single retrieval-inference loop, and thus cannot ensure the top priority of hard constraints at the system level. At the same time, this paper distinguishes between market information and regulatory constraints and assigns different processing paths to each in its dynamic knowledge hierarchy.

2.2 Code-enhanced reasoning and rule execution.

Another type of related work aims to improve the reliability of the reasoning process by code augmentation or rule enforcement. CoT and its variations aim to enhance the step-by-step reasoning capability of large language models for complex tasks by adding intermediate steps; other methods include PoT and PAL, which externalise parts of the reasoning process as executable programs and have achieved good results in numerical computation and logical reasoning. A general feature of the above ways is that they break down generation and execution into several steps to enhance the determinism of individual steps. However, these methods mainly focus on computational correctness, program-assisted solving or single-step verifiability, and generally assume that the task objectives can be expressed as well-defined computational or program-execution problems. The first problem with the subjects of stock trading regulation is that the rules are in natural language and cannot be easily converted into machine-readable constraints for the whole decision-making process. This paper builds executable rules into a structured constraint logic and has enabled these rules to be used more openly in the assessment of candidate behaviour.

2.3 Self-Correction in Multi-Agent Systems

Another direction of research for long-chain reasoning in complex tasks is multi-agent and self-correction methods. ReAct and AutoGPT extend the flexibility of the system for multi-step tasks by alternating between reasoning and execution; at the same time, Self-Refine and Reflexion enhance the reliability of results through iterative feedback, self-correction or multiple rounds of reflection. The above ways are to promote learning and adjustment by providing feedback at different stages. For high-risk stock trading, even self-reflection, negotiation or self-correction is insufficient to avoid building up early biases over a long period of decision-making; stable guarantees under strict compliance constraints are thus particularly difficult to achieve. Based on the idea of multiple steps in decision-making and process control, three audit mechanisms have been proposed at the three stages of knowledge filtering, constraint enforcement and final decision-making in this paper. Thus, the errors will not occur, and the system will maintain an abnormal state.

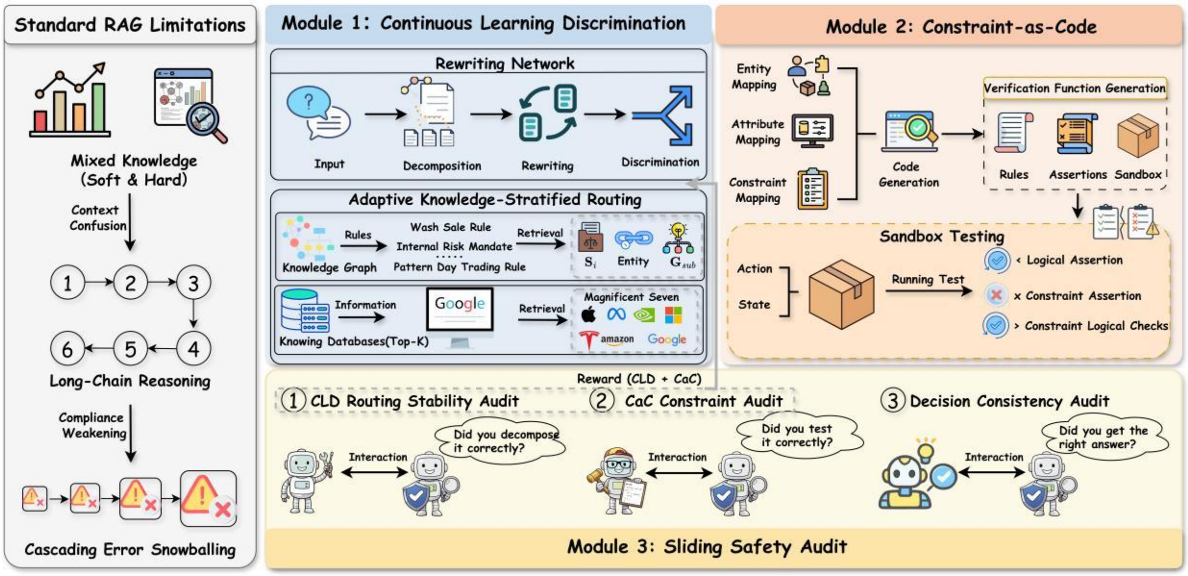


Figure 1: FinGuard Architecture. CLD semantically decomposes the input query, and CaC turns the compliance knowledge into executable constraints. The above constraints direct the generation process, and SSA will perform repeated adversarial checks and consistency examinations on the resulting decision.

3 FinGuard for Compliant Stock Trading

3.1 Introduction

Input queries in high-risk stock trading typically include market conditions, trading objectives, and rule constraints; therefore, the system not only needs to comprehend dynamic information but also generate compliant actions that can be executed in practice. Given a user query **Q**, a market-related document repository D_{vec}, and a compliance knowledge graph G_{KG}, the objective is to generate a candidate decision **Y** that leverages dynamic market information while consistently satisfying rigid compliance constraints.

The three modules of FinGuard are continuous learning discrimination (CLD), constraint-as-code (CaC), and sliding safety audit (SSA) (as shown in Figure 1). CLD first semantically decomposes the initial query and then routes different sub-intentions to the two channels: soft knowledge and hard knowledge. The former presents the market environment and raises problems; the latter derives executable rule constraints. CaC then compiles the rule subgraphs from the hard knowledge channel into executable assertion logic and performs deterministic verification of the candidate decisions. Finally, SSA will conduct a sliding audit of the three core links in the knowledge-routing-constraint-execution-final-decision pathway to avoid the accumulation of initial errors in a long chain of reasoning. Only when the candidate action meets all hard constraints and passes the stage-wise audits will the system output the final decision.

3.2 Soft and Hard Knowledge

Soft knowledge is time-varying market information that is probabilistic and context-dependent, such as price trends, news events, analytical opinions, trading emotions, etc. This kind of knowledge helps the model understand the current market environment and determine opportunities and risks; therefore, similarity-based retrieval of a vectorised document repository is more suitable. Hard knowledge, on the other hand, refers to information with standardised semantics that is non-negotiable and directly defines the space of possible actions, such as regulatory provisions, risk control thresholds, position limits and trading behaviour constraints. This type of knowledge sets the permissible range for decision-making and thus needs to be included in the reasoning as structured graphs and executable logic.

Based on this framework, FinGuard uses a vector database $\boldsymbol{D_{vec}}$ to represent soft knowledge and a compliance knowledge graph $\boldsymbol{G_{KG}}$ to represent hard knowledge. The former supports context awareness and strategy generation, while the latter supports constraint resolution and compliance verification.

3.3 Continuous Learning and Discrimination

Financial queries generally combine transaction intentions, market signals and regulatory requirements, and the same expression may have different semantic roles in different contexts. For instance, some text can be required by the regulation in one case but is only a risk warning in another. If the query classification is only based on static templates or manual rules, it will be difficult to distinguish between dynamic information and fixed constraints in a complex task reliably. Therefore, CLD is proposed in this paper for adaptive query decomposition and knowledge routing at runtime.

Given an initial query **Q**, CLD first employs the rewrite network M_{ϕ} to perform semantic decomposition on it, generating a set of sub-fragments:

$$S=\mathrm{M}_{\phi}(Q)=\{s_1,s_2,\ldots,s_n\}$$

Each sub-fragment is a relatively complete decision intention that has a more singular semantic focus. On the one hand, it reduces the ambiguity of long sentences and complex commands and implicit financial terms; at the same time, it establishes clear semantic boundaries for the processing of heterogeneous knowledge later.

After obtaining the set of sub-fragments, CLD learns a binary routing function $\boldsymbol{g}(\boldsymbol{s_i})\in\{0,1\}$ for each fragment. When $\boldsymbol{g}(\boldsymbol{s_i})=0$, the fragment is classified as a soft knowledge query and sent to the vector store for ***Top-k*** retrieval; When $\boldsymbol{g}(\boldsymbol{s_i})=1$, the fragment is classified as a hard knowledge query and mapped to the knowledge graph via entity links to extract relevant subgraphs.

$$\boldsymbol{K}=\bigcup_{s_i:g(s_i)=0}\textbf{\textit{Top-k}}\,(s_i,D_{vec})\cup\bigcup_{s_i:g(s_i)=1}\textbf{\textit{Subgrap}}\mathrm{h}\,(s_i,G_{KG})$$

$\boldsymbol{K}$contains both market reference information and rule constraint information; however, these two components have been separated within the system and are no longer combined during the retrieval phase.

As the results of CLD directly affect the subsequent constraint compilation and security auditing, it will be considered a continuous learning process in this paper. The compilation successful flag from CaC and the audit pass signal from SSA are used as downstream feedback to define the objective function:

$$J(\theta)=E_{\tau\sim\pi_\theta}\left[\alpha\cdot I_{\mathrm{compile}}(\tau)+\beta\cdot\sigma_{\mathrm{audit}}(\tau)\right]$$

Here, π_θ denotes the CLD’s routing policy, $I_{compile}(\tau)$ indicates whether the current routing outcome supports subsequent rule compilation, and $\sigma_{audit}(\tau)$ indicates the degree to which the corresponding inference chain passes the audit phase. Routing strategies that consistently produce compilable and auditable decision chains are reinforced, while those that frequently cause compilation failures or audit rejections are suppressed. CLD continuously adjusts the soft/hard decision boundary through task feedback, thereby better adapting to the complex and dynamic semantic structures found in high-risk financial scenarios.

3.4 Constraint-as-Code

If the compliance rules are only expressed in prompts, then the constraints cannot be enforced consistently at all times during decision-making. Weak long-range reasoning may be the cause, or the constraints in a particular context have been ignored despite the reasons being locally reasonable. To the extent that the above statement is correct, CaC is proposed here; it converts executable compliance rules from text into deterministic logic and allows them to participate in decision-making as an independent adjudicative mechanism.

Based on the hard knowledge subgraphs provided by CLD, this paper builds rule subgraphs:

$$G_{sub}=\{(\mathrm{h},r,t)\mid \mathrm{h},t\in \mathrm{E},\ r\in \mathrm{R}\}$$

This paper introduces a dedicated Coder Agent for compiling it into the Python logic space:

$$\mathrm{F}_{\mathrm{compile}}:G_{sub}\rightarrow C$$

Here, Cdenotes a set of executable constraint functions. The compilation process follows three basic mappings: entity mapping converts entity nodes in the graph into constraint objects or variables; attribute mapping converts attributes such as numerical thresholds and discrete states into comparable parameters; and constraint mapping converts normative relationships into Boolean assertions. Through this process, rules originally expressed in natural language and graph structures are transcribed into executable assertion logic. For any candidate decision Y, CaC defines a unified constraint operator:

$$\Phi(\boldsymbol{Y})=\bigwedge_{c_i\in C_{derived}} \boldsymbol{I}(\boldsymbol{Eval}(\boldsymbol{c_i},\boldsymbol{Y})=\text{True})$$

A candidate action will only be deemed executable if all derived constraints return True; otherwise, it will be rejected and returned to the upstream process for regeneration. To guarantee the determinism of execution, all compiled logic code operates in a closed sandbox and is thus decoupled from natural language generation.

CaC converts rules into logic that needs to be executed. Therefore, compliance assessments will no longer be based on an ad hoc application of the model in various situations but will be objective and reliable. Not all the norms and standards are translated into logical code. If the rules are too general or have no clear operating definitions, or if there is a contradiction among them, they will be stored in the soft knowledge channel for reference. Therefore, no new rule bias will be introduced by the incorrect compilation, and CaC will still work normally in the actual rule environment with determinism.

3.5 Sliding Safety Audit

Errors in high-risk stock trading do not always occur at the final output stage. Biases can occur at any time in the knowledge routing, rule compilation or intermediate inference nodes, and gradually spread along the decision path. To solve this problem, this paper proposes SSA; it uses sliding audit windows at various points in the inference chain to conduct segmented inspections of intermediate states and initiates retries or rollbacks upon the occurrence of a local failure.

The three stages of the sequential advance of the SSA audit window are CLD routing stability audit, CaC constraint boundary audit, and final decision consistency audit. The window will only move to the next stage after successfully completing the current stage, and if there is an inconsistency, it will return to the corresponding module for local correction. As a result, SSA serves as a control mechanism for processes that can enhance the stability of long-chain reasoning proactively.

3.5.1 CLD Routing Stability Audit

To verify whether the CLD’s routing results remain stable under semantic disruptions, SSA first generates adversarial disruption samples $\boldsymbol{s_i}'=\boldsymbol{s_i}\oplus\boldsymbol{\delta}$ for each sub-sequence $\boldsymbol{s_i}$, and compares the differences in routing distributions between the original input and the disrupted input:

$$\boldsymbol{T}_{\text{retry}}(\boldsymbol{s_i})=\boldsymbol{I}(\boldsymbol{D}_{\text{KL}}(\boldsymbol{p}(\boldsymbol{z}\mid \boldsymbol{s_i})\parallel \boldsymbol{p}(\boldsymbol{z}\mid \boldsymbol{s_i}'))>\boldsymbol{\tau_{stable}})$$

If the above conditions are met, it means that the semantic boundaries of the current sub-fragment are unstable, and thus the system will return it to the CLD for re-decomposition and re-routing. Help avoid misrouting of ambiguous expressions to the soft or hard channels, thus reducing the risk of subsequent rule compilation failures or decision bias.

3.5.2 CaC Constraint Boundary Audit

Many rule violations occur in the high-risk area close to the compliance limit. SSA generates a set of boundary test cases close to each constraint for the constraint code generated by CaC.

$$\boldsymbol{Y_{edge}}=\left\{\boldsymbol{y}\mid \exists\, \boldsymbol{c}\in \boldsymbol{C_{derived}},\ \min_{\boldsymbol{y_b}\in\partial\Omega(\boldsymbol{c})}\|\boldsymbol{y}-\boldsymbol{y_b}\|_2<\epsilon\right\}$$

The current code module is considered to have passed the audit only if the constraint solver Φ agrees with the logical evaluation results for all boundary samples; otherwise, the system will trigger a constraint recompilation or discard the current compilation result.

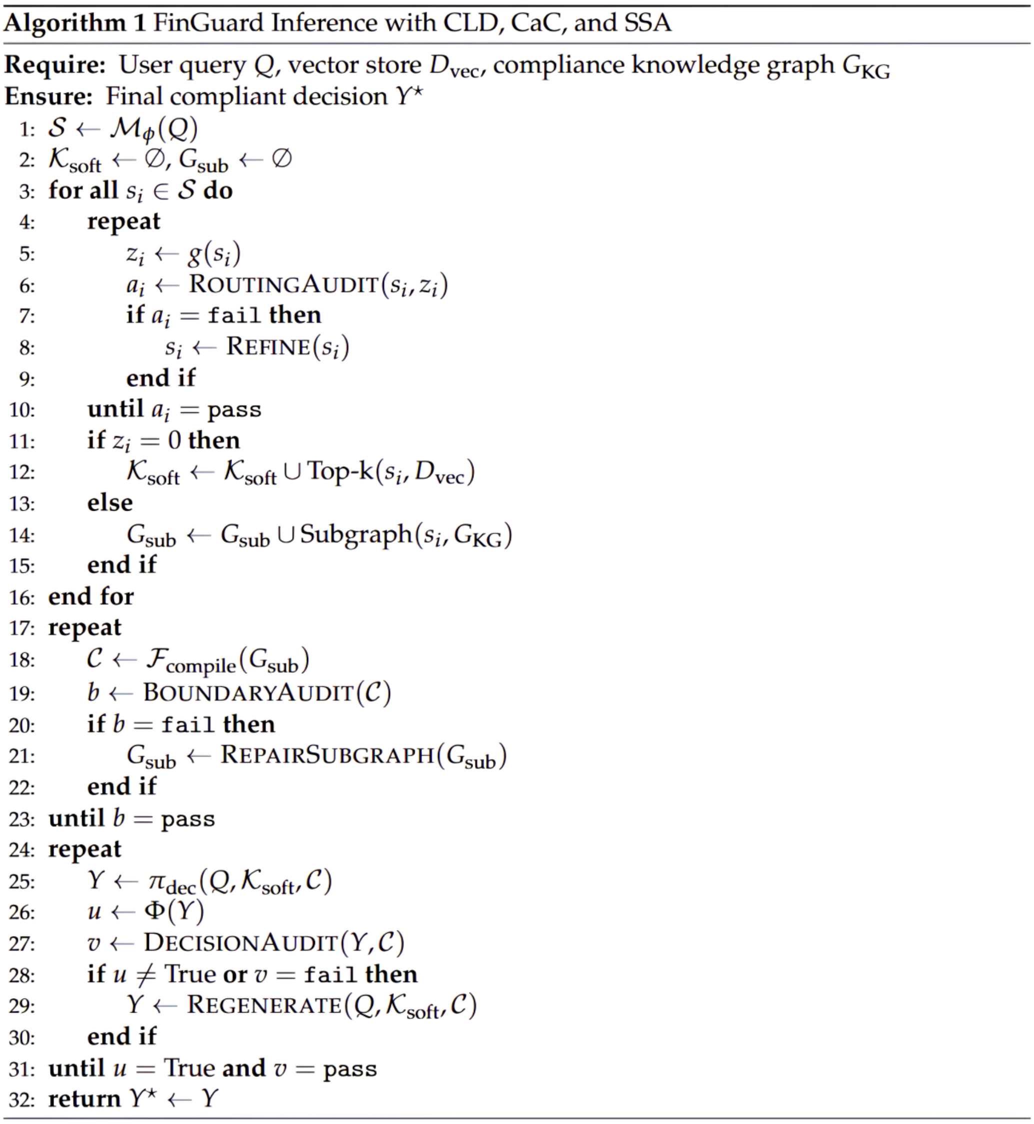
3.5.3 Decision Consistency Audit

During the final decision-making stage, SSA uses the counterfactual hypothesis generator $\boldsymbol{G_{cf}}$ to construct a set of what-if scenariosH that are adjacent to the current situation but more challenging, and requires the candidate strategies to still satisfy the hard constraints under all these scenarios:

$$\forall h \in H, \quad \Phi(\boldsymbol{\pi}_{\text{dec}}(\boldsymbol{Q} \oplus h))=\text{True}$$

If a candidate action is compliant only in the current context but violates key constraints under reasonable perturbation, then the system rejects the action and triggers regeneration.

Algorithm 1 is used to show how CLD, CaC and SSA work together in the inference process for FinGuard.



Algorithm 1 FinGuard Inference with CLD, CaC, and SSA

```
Require: User query $Q$, vector store $D_{\text{vec}}$, compliance knowledge graph $G_{\text{KG}}$
Ensure: Final compliant decision $Y^{\star}$
$\mathcal{S} \leftarrow \mathcal{M}_{\phi}(Q)$
$\mathcal{K}_{\text{soft}} \leftarrow \emptyset$, $G_{\text{sub}} \leftarrow \emptyset$
for all $s_i \in \mathcal{S}$ do
    repeat
        $z_i \leftarrow g(s_i)$
        $a_i \leftarrow \textsc{RoutingAudit}(s_i, z_i)$
        if $a_i = \texttt{fail}$ then
            $s_i \leftarrow \textsc{Refine}(s_i)$
        end if
    until $a_i = \texttt{pass}$
    if $z_i = 0$ then
        $\mathcal{K}_{\text{soft}} \leftarrow \mathcal{K}_{\text{soft}} \cup \text{Top-k}(s_i, D_{\text{vec}})$
    else
        $G_{\text{sub}} \leftarrow G_{\text{sub}} \cup \text{Subgraph}(s_i, G_{\text{KG}})$
    end if
end for
repeat
    $\mathcal{C} \leftarrow \mathcal{F}_{\text{compile}}(G_{\text{sub}})$
    $b \leftarrow \textsc{BoundaryAudit}(\mathcal{C})$
    if $b = \texttt{fail}$ then
        $G_{\text{sub}} \leftarrow \textsc{RepairSubgraph}(G_{\text{sub}})$
    end if
until $b = \texttt{pass}$
repeat
    $Y \leftarrow \pi_{\text{dec}}(Q, \mathcal{K}_{\text{soft}}, \mathcal{C})$
    $u \leftarrow \Phi(Y)$
    $v \leftarrow \textsc{DecisionAudit}(Y, \mathcal{C})$
    if $u \neq \text{True}$ or $v = \texttt{fail}$ then
        $Y \leftarrow \textsc{Regenerate}(Q, \mathcal{K}_{\text{soft}}, \mathcal{C})$
    end if
until $u = \text{True}$ and $v = \texttt{pass}$
return $Y^{\star} \leftarrow Y$
```

4 Experimental Evaluation

We will conduct some controlled experiments to evaluate FinGuard's compliance and long-term stability in a high-risk stock trading environment in this section. Evaluation of Reliability Issues Caused by Long-Chain Reasoning and Rule-Dense Constraints. We aim to answer the following three questions: RQ1, can FinGuard maintain stable decision performance under compliance constraints at all levels of transaction time granularity; RQ2, can it prevent the accumulation of errors and hidden violations in dynamic long-chain decision-making processes; and RQ3, what roles do the core modules of FinGuard play in alleviating the problems mentioned above?

4.1 Experimental Arrangement

Datasets and Implementation. To evaluate FinGuard's performance in rule intensivestock trading with high error costs, we constructed a controlled stock trading dataset. We constructed the dataset using stocks from seven technology companies—the Magnificent Seven (e.g., AAPL and NVDA)—collected from Investing.com from 2022 to 2024, yielding 1,764 trading days (252 × 7). This setup provides strong experimental control while facilitating analysis of stability differences among methods under long-term operation and compliance constraints.

At the same time, we have designed 100 investment queries and incorporated 60 core financial hard rules to build decision-making tasks for position, risk exposure and trading behaviour. The model needs to generate a trading decision that adheres to all the hard rules, and finally, the return is used to judge how good this decision is in a compliant way across different methods. All the evaluation used GPT-4o as the base model for fairness. A uniform generation temperature was used, and all methods ran under the same budget constraints to reduce the effect of randomness. To study the behavioural differences at various times of the decision-making process further, three levels of aggregation were set up: daily (T=1), weekly (T=7), and monthly (T=28). T is the time interval from decision generation to performance evaluation here. A model makes decisions every T trading days, and then the returns are determined by the market performance in that cycle to compare the results of decision-making at different time scales.

Baselines. FinGuard is compared with the six typical baseline methods in the paradigm of RAG, stepwise reasoning and agent decision-making for current large language models: (1) ReAct is a representative agentic RAG framework that conducts multi-step planning and tool invocation by alternating between reasoning and action; (2) CoT enhances the quality of intermediate reasoning in complex scenarios by adding chain-of-thought prompts to RAG; (3) Standard RAG uses Top-k retrieval combined with generative models for decision-making; (4) Self-Refine improves model output through an iterative self-feedback mechanism; (5) PoT offloads partial reasoning to an external Python interpreter and shows stability in numerical computation and logical deduction tasks; (6) Zero-shot is a pure model benchmark that does not use external knowledge.

Although the above baselines have certain strengths in reasoning ability or generation consistency, their adherence to constraints relies solely on prompt design or the model's inherent knowledge; there is no systematic deterministic execution mechanism. The above two are not the same, nor do they follow the same path.

Metrics. The three indicators used to measure the behaviour of models that carry a high risk of stock trading are accuracy, compliance and return. Accuracy is the agreement between the oracle trading actions under full-rule compliance and the realised market results; Compliance refers to the proportion of decisions generated by the model that meet the hard constraints, thereby assessing how closely it adheres to the non-negotiable rules; Raw Return is the accumulated return before applying risk penalties, indicating the model's profit potential under ideal circumstances.

Given that the violations in real life often have hidden risks and latent costs, we will introduce a penalty mechanism based on the frequency of the violations. Adjusted Return is used to account for compliance and returned values. The discount factor of the return is as follows:

$$\boldsymbol{d} = \max\left(\boldsymbol{d}_{\min}\ ,\ 1 - \boldsymbol{v} \cdot \boldsymbol{\delta}\right)$$

where d represents the discount factor for returns, v denotes the number of violations, and δ is the discount rate per violation. When a trade generates positive returns, the discount mechanism reduces the returns proportionally to the number of violations. When a trade incurs losses, this discount effectively magnifies the losses, thereby intensifying penalties for high-risk behavior. Simultaneously, a lower bound $d_{\min}$ is set to cap the severity of penalties (maximum penalty of 50%) to prevent assessment instability.

Table 1: Compare the performance of FinGuard with multiple baseline methods at different time granularities and use zero-shot serving as the unified reference. Red annotations show the largest improvement over Zero-shot, and subscripts are used to indicate the exact gain value.

Time	Model	Accuracy	Raw Return	Adj Return	Compliance
	ReAct Agent	$37\%_{+10\%}$	$+0.46\%_{+0.07\%}$	$+0.44\%_{+0.07\%}$	$91\%_{+8\%}$
	CoT-RAG	$36\%_{+9\%}$	$+0.44\%_{+0.05\%}$	$+0.42\%_{+0.05\%}$	$88\%_{+5\%}$
	Standard RAG	$33\%_{+6\%}$	$+0.42\%_{+0.03\%}$	$+0.40\%_{+0.03\%}$	$90\%_{+7\%}$
Daily	Self-Refine	$34\%_{+7\%}$	$+0.45\%_{+0.06\%}$	$+0.44\%_{+0.07\%}$	$92\%_{+9\%}$
	PoT	$35\%_{+8\%}$	$+0.44\%_{+0.05\%}$	$+0.44\%_{+0.07\%}$	$90\%_{+7\%}$
	Zero-shot	27%	+0.39%	+0.37%	83%
	FinGuard	$39\%_{+12\%}$	$+0.48\%_{+0.09\%}$	$+0.49\%_{+0.12\%}$	$97\%_{+14\%}$
	ReAct Agent	$36\%_{+11\%}$	$+0.45\%_{+0.09\%}$	$+0.43\%_{+0.08\%}$	$89\%_{+9\%}$
	CoT-RAG	$34\%_{+9\%}$	$+0.44\%_{+0.08\%}$	$+0.43\%_{+0.08\%}$	$88\%_{+8\%}$
	Standard RAG	$32\%_{+7\%}$	$+0.41\%_{+0.05\%}$	$+0.41\%_{+0.06\%}$	$89\%_{+9\%}$
Weekly	Self-Refine	$34\%_{+9\%}$	$+0.44\%_{+0.08\%}$	$+0.42\%_{+0.07\%}$	$86\%_{+6\%}$
	PoT	$33\%_{+8\%}$	$+0.42\%_{+0.06\%}$	$+0.40\%_{+0.05\%}$	$88\%_{+8\%}$
	Zero-shot	25%	+0.36%	+0.35%	80%
	FinGuard	$38\%_{+13\%}$	$+0.47\%_{+0.11\%}$	$+0.47\%_{+0.12\%}$	$97\%_{+17\%}$
	ReAct Agent	$36\%_{+16\%}$	$+0.50\%_{+0.16\%}$	$+0.47\%_{+0.17\%}$	$87\%_{+11\%}$
	CoT-RAG	$33\%_{+13\%}$	$+0.45\%_{+0.11\%}$	$+0.42\%_{+0.12\%}$	$86\%_{+10\%}$
	Standard RAG	$31\%_{+11\%}$	$+0.42\%_{+0.08\%}$	$+0.40\%_{+0.10\%}$	$84\%_{+8\%}$
Monthly	Self-Refine	$33\%_{+13\%}$	$+0.43\%_{+0.09\%}$	$+0.40\%_{+0.10\%}$	$86\%_{+10\%}$
	PoT	$30\%_{+10\%}$	$+0.40\%_{+0.06\%}$	$+0.39\%_{+0.09\%}$	$85\%_{+9\%}$
	Zero-shot	20%	+0.34%	+0.30%	76%
	FinGuard	$40\%_{+20\%}$	$+0.48\%_{+0.14\%}$	$+0.50\%_{+0.20\%}$	$95\%_{+19\%}$

4.2 Main Results

Overall Performance Comparison. As shown in Table 1, the overall results of all methods for Accuracy, Compliance and Return are shown at different time granularities (Daily/Weekly/ Monthly). FinGuard has achieved the best or near-best performance in terms of adjusted return and compliance across all the time granularities evaluated, and is thus relatively stable in balancing profit generation with rule constraints. FinGuard generally achieved a much higher compliance rate than the general-purpose agent methods ReAct and Self-Refine in all three time granularity tasks. Although ReAct had a higher raw return in the Monthly scenario, its relatively low compliance rate did not lead to an increase in adjusted return after adding Adj Return. Therefore, the higher returns have been achieved by taking greater risks.

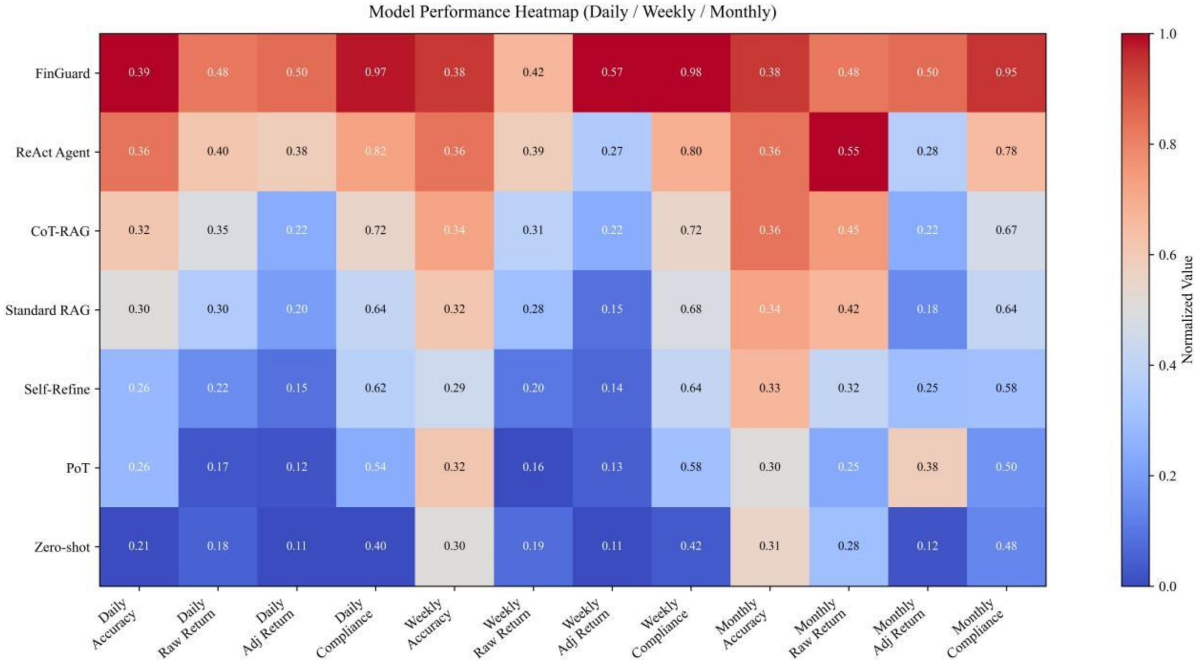


Figure 2: Heatmap of all three methods at different times of aggregation (daily/weekly/ monthly), with colours indicating normalised metric values.

As shown in Figure 2, most of the baseline methods have experienced a considerable drop in Adjusted Return compared with Raw Return after the introduction of penalty mechanisms. It can be seen that, without limitations, models are more likely to violate the rules for short-term benefits and thus suffer larger losses after risk adjustment. FinGuard has been doing reasonably well and within the range of compliance so far. FinGuard still has a good compliance rate and relatively stable risk-adjusted returns under the longer decision cycles of weekly and monthly. Therefore, it can better prevent the spread of errors in extended chains of reasoning and information accumulation. It will also support the function of CLD in knowledge routing under special circumstances and ensure the long-term stability of the system.

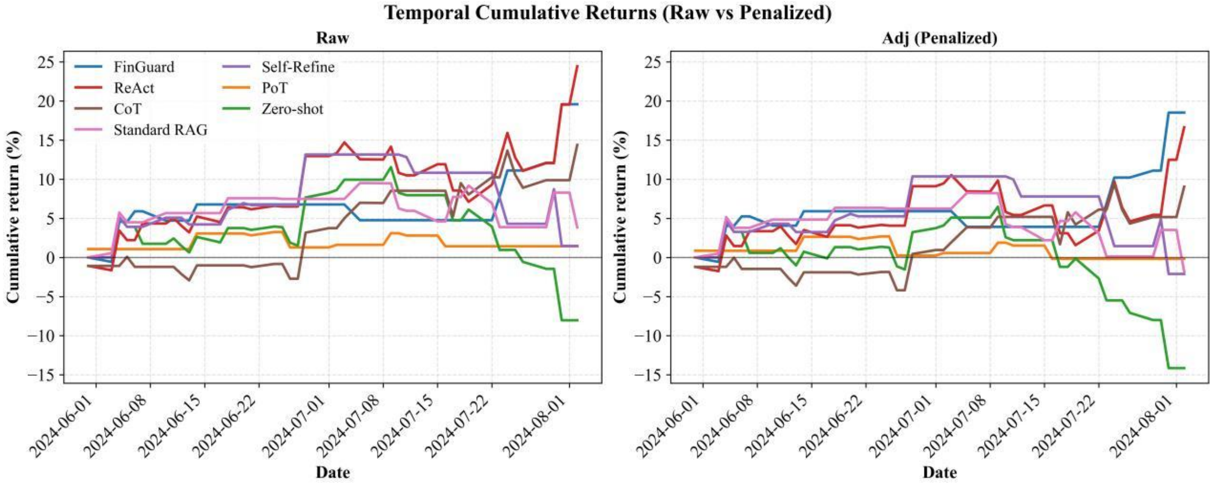


Figure 3: Cumulative return curves for all methods over the past 44 trading days are shown, and raw returns (Raw) are compared with risk-adjusted returns (Adj) after adding violation penalties.

4.3 Temporal Evaluation and Robustness

To check the stability of the system under extended operation, a 44-day time-series stress test was conducted and both raw and penalty-adjusted returns were recorded.

As shown in Figure 3, most of the basic methods have achieved a short-term gain in the Raw mode, but their cumulative return curves are gradually unstable after a long period of trading. After the introduction of violation penalties, several baselines have shown a clear return degradation; thus, it can be concluded that some of their raw performance is achieved through decisions that are less robust under compliance constraints. FinGuard has a lower trajectory of cumulative returns and has been relatively stable; thus, its profitability is less affected by rule violations or high-risk behaviour.

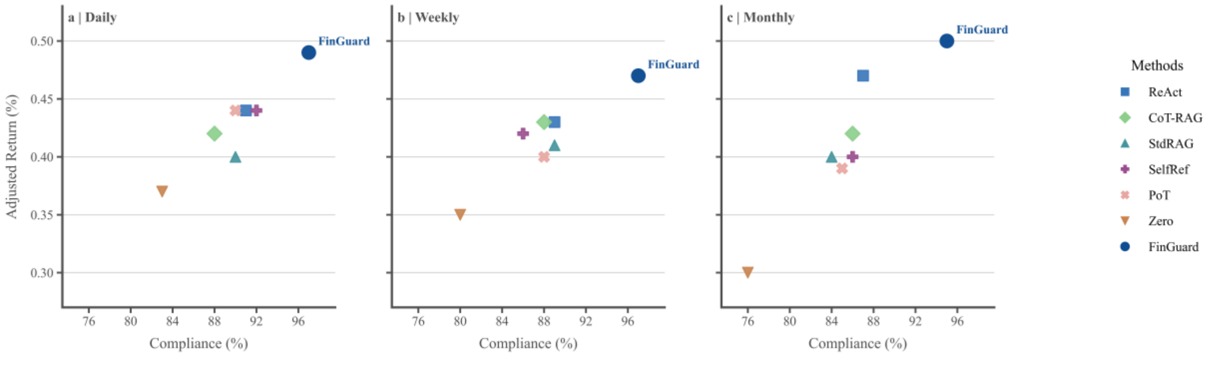


Figure 4: Compliance-Adjusted Return Comparison Across Daily, Weekly and Monthly Horizons. The x-axis is compliance, and the y-axis is adjusted return.

Figure 4 also shows the relationship between compliance and adjusted return for the Daily, Weekly and Monthly decision horizons. FinGuard is continuously in the upper-right region, so it has both high compliance and strong risk-adjusted returns. The first few years are more representative of this phenomenon; later, this loss of compliance or return will become more pronounced. Therefore, it can be seen that FinGuard is relatively stable in the event of extended decision intervals and increased error accumulation.

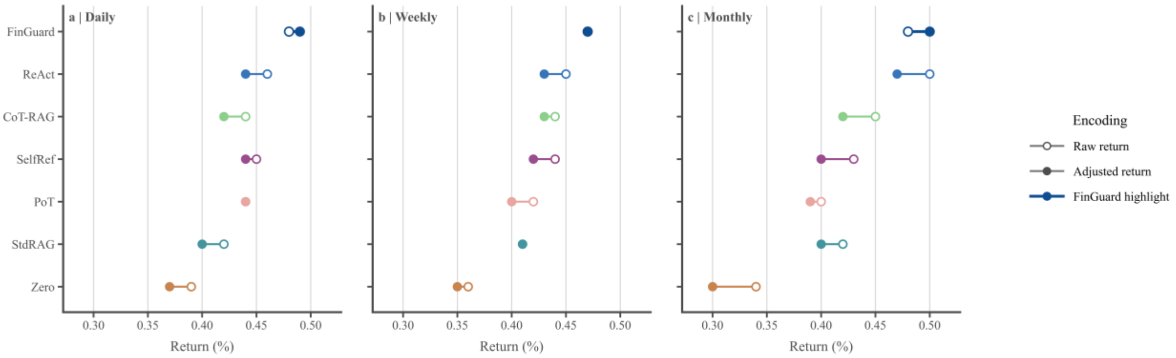


Figure 5: Comparison of Raw Returns and Adjusted Returns at Different Decision Horizons. The distance of the two markers is the change in return after the imposition of compliance penalties.

Figure 6 shows the Raw Return and Adj Return of all methods. Most of the baselines show that, after adjustments, their returns have fallen compared to the raw returns; thus, the impact of compliance penalties has reduced the initial gains. FinGuard has a very small gap between Raw and Adj returns across all three horizons, on the other hand. It can be seen from the above that its decisions are less sensitive to penalty adjustment and that most of the return performance is realised within the compliance area.

Table 2 shows the quantitative summary of the above. FinGuard was at the top with an average compliance rate of 96.3%, a relatively high average adjusted return of +0.487%, and an almost non-existent average signed penalty gap. It can be seen from the above that, compared with the base methods, FinGuard has a higher adjusted performance and a lower dependency on returns that would be eliminated after compliance penalties are taken into account. FinGuard generally has relatively stable temporal robustness results, and both continuous and cross-horizon trading can achieve a good balance of profit and compliance.

Table 2: Penalty Sensitivity and Cross-Horizon Robustness of FinGuard and Baseline Methods. The signed penalty gap is calculated as Raw Return - Adj Return; a smaller value indicates that the gains excluded due to compliance penalties have a relatively lower impact, while a higher average compliance and average adjusted return suggest that the constrained trading is less severely impaired.

Method	Daily Gap↓	Weekly Gap↓	Monthly Gap↓	Avg. Gap↓	Avg. Compliance↑	Avg. Adj. Return↑
ReAct Agent	+0.02%	+0.02%	+0.03%	+0.023%	89.0%	+0.447%
CoT-RAG	+0.02%	+0.01%	+0.03%	+0.020%	87.3%	+0.423%
Standard RAG	+0.02%	+0.00%	+0.02%	+0.013%	87.7%	+0.403%
Self-Refine	+0.01%	+0.02%	+0.03%	+0.020%	88.0%	+0.420%
PoT	+0.00%	+0.02%	+0.01%	+0.010%	87.7%	+0.410%
Zero-shot	+0.02%	+0.01%	+0.04%	+0.023%	79.7%	+0.340%
FinGuard	-0.01%	+0.00%	-0.02%	-0.010%	96.3%	+0.487%

4.4 Ablation Study

To confirm that all core modules in FinGuard are necessary, CLD, CaC and SSA were removed individually without changing any of the other training and evaluation parameters, and then experiments were repeated at various time scales (Figure 6). As shown in Table 3, the performance of the system has dropped considerably when any single module is excluded. Thus, FinGuard's observed advantages are not driven by a higher degree of risk-taking but rather by the combined effects of knowledge routing, constraint adjudication and reasoning audit.

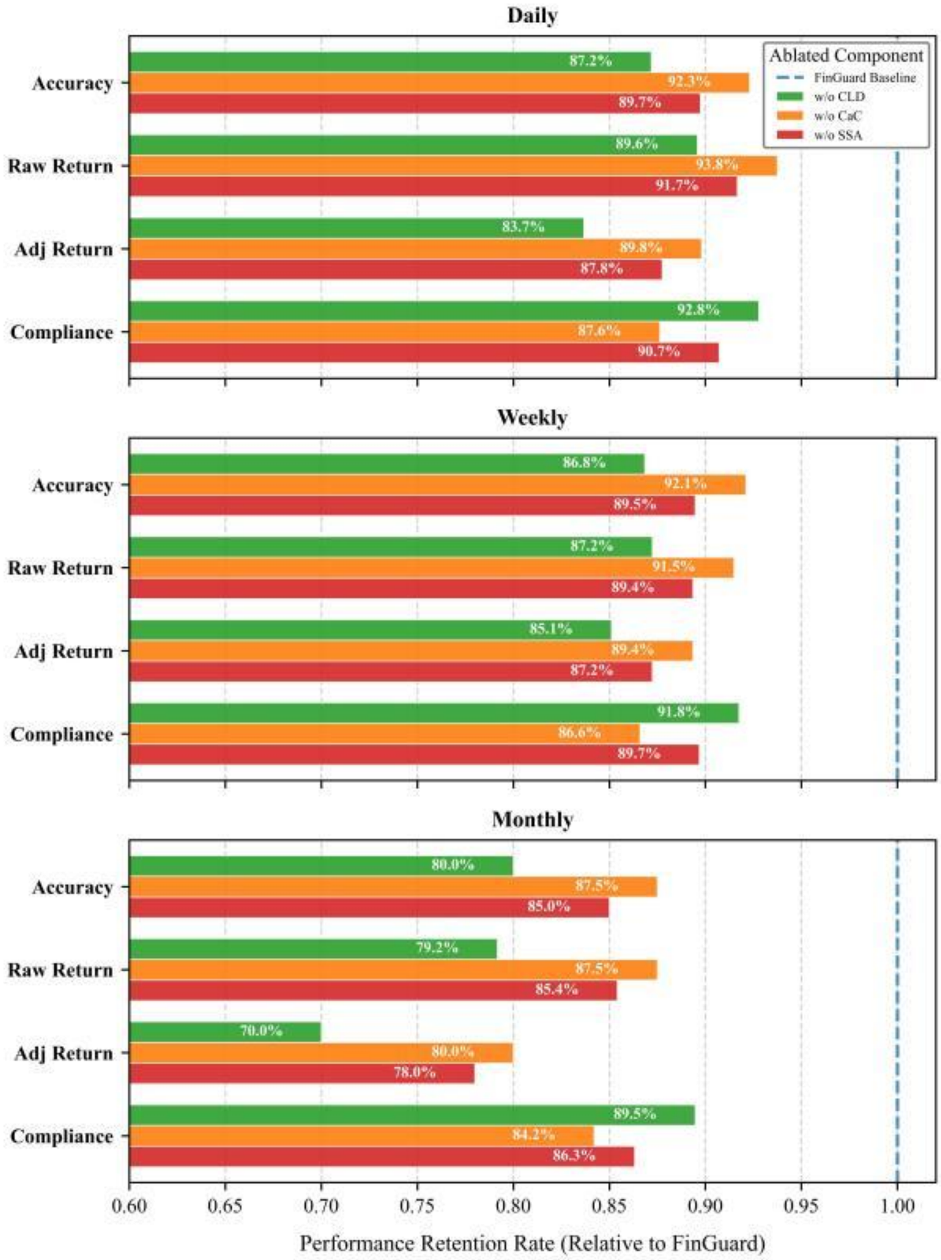


Figure 6: Impact of Component Removal on Model Performance at Various Granularities.

4.4.1 CLD is suitable for supporting medium- to long-term decision-making.

CLD is removed from the model, and it can no longer distinguish between fluctuations in market data and fixed-rule constraints across multiple decision cycles; thus, both the rate of compliance and risk-adjusted returns decrease simultaneously. It can be seen that when rules and contexts are highly interdependent, a lack of an effective knowledge-separation mechanism will further increase reasoning confusion and make compliance constraints more easily hidden in the market noise.

4.4.2 CaC Ensures Compliance and Stable Returns

Without CaC, the model may maintain some initial returns in appearance; however, compliance will be significantly reduced, and as a result, systemic erosion of returns will occur after adding risk penalties. Therefore, it can be seen that generative reasoning alone may not be suitable for consistently enforcing strict rules over an extended period in decision-making, and thus explicit constraint adjudication is needed to reduce violations.

4.4.3 SSA reduces error propagation in long-chain reasoning

After excluding SSA, the model shows a decreasing trend in stability across different levels of temporal granularity; thus, relying solely on the final output will fail to prevent the accumulation of early biases through repeated decision cycles. A continuous, staggered-off auditing mechanism needs to be established to ensure the reliability of long-term decision-making behaviour.

Table 3: Ablation results of FinGuard after removing different core modules are shown.

Time	Model Variant	Accuracy	Raw Return	Adj Return	Compliance
Daily	w/o CLD	34%↓5%	+0.43%↓0.05%	+0.41%↓0.08%	90%↓7%
	w/o CaC	36%↓3%	+0.45%↓0.03%	+0.44%↓0.05%	85%↓12%
	w/o SSA	35%↓4%	+0.44%↓0.04%	+0.43%↓0.06%	88%↓9%
	FinGuard	39%base	+0.48%base	+0.49%base	97%base
Weekly	w/o CLD	33%↓5%	+0.41%↓0.06%	+0.40%↓0.07%	89%↓8%
	w/o CaC	35%↓3%	+0.43%↓0.04%	+0.42%↓0.05%	84%↓13%
	w/o SSA	34%↓4%	0.42%↓0.05%	+0.41%↓0.06%	87%↓10%
	FinGuard	38%base	0.47%base	+0.47%base	97%base
Monthly	w/o CLD	32%↓8%	+0.38%↓0.10%	+0.35%↓0.15%	85%↓10%
	w/o CaC	35%↓5%	+0.42%↓0.06%	+0.40%↓0.10%	80%↓15%
	w/o SSA	34%↓6%	+0.41%↓0.07%	+0.39%↓0.11%	82%↓13%
	FinGuard	40%base	+0.48%base	+0.50%base	95%base

4.5 Discussion

Mechanism. FinGuard's advantages stem from the combined effect of knowledge routing, deterministic constraint adjudication and process auditing. Knowledge routing reduces the mutual interference of market signals and rigid rules in a unified retrieval-inference loop; CaC converts executable rules into deterministic adjudication logic; and SSA moves the review process to the interior of the inference process to make local corrections before biases spread. Based on the results of the ablation experiment, when these mechanisms are removed, the system's compliance rate, adjusted returns and long-term stability all show a decline.

Trad-off. The results of this study indicate that the system of high-risk financial decisions should consider compliance, long-term stability and risk-adjusted return performance simultaneously. Although some of the basic methods have achieved relatively high gross income, they have not accounted for compliance expenses; therefore, their actual net earnings after considering these costs will be lower. FinGuard has a lower compliance rate, a gentler cumulative return curve, and more volatile adjusted returns at all times during the evaluation period. Based on the above results, it can be seen that in environments with strict rules and a high cost of errors, the goal should not be to maximise short-term returns but rather to make stable decisions within the constraints.

Implications. For stock trading tasks, controllability, auditability and compliance reliability should not be viewed as mere results of profit optimisation but should be included in the system goals themselves. Therefore, the structure of FinGuard does not have an unnecessary multiple layers; rather, different sources of risk are allocated to different processing levels to improve analyzability and intervenability. Strengthened control will be introduced for high-performance, stable, and compliant-with-regulations systems that require a certain level of security.

Limitations. The above have been covered. Firstly, the present verification is based on a controlled stock trading environment and cannot yet be compared with market shocks, execution delays and other changes in real online trading systems. Second, the scope of assets, query sets and rule sets used in the experiments is relatively narrow; generalisation across different markets, asset classes and under more complex regulatory systems still needs to be tested further. Thirdly, the efficiency of CaC depends on whether there is some structure and compilation capability in the rules; there are still deficiencies in the application of it to ambiguous, contradictory or frequently altered regulations. Future work will extend to more complex real-world market environments, add dynamic rule updates and a human-in-the-loop review mechanism, and systematically study the trade-offs among audit overhead, response latency and deployment costs.

5 Conclusion

Systematically analyze the reliability problems of Agentic RAG in a high-risk stock trading environment over an extended period under strict compliance requirements. Identify the causes of norm erosion and error proliferation in long-chain reasoning under the current system. We believe that, in particular, the structural integration of market soft knowledge and compliance hard knowledge into the probabilistic retrieval and reasoning process is the main reason for the gradual weakening of compliance constraints on decision-making. Based on the above observation, we put forward the FinGuard framework. Improve compliance enforcement and decision stability by adaptive knowledge stratification and routing, deterministic adjudication of enforceable constraints, and continuous auditing of long-chain reasoning. Based on the experimental results, FinGuard obtained the best or near-best risk-adjusted returns in the daily, weekly, and monthly scenarios and maintained a compliance rate of 95% to 97%; therefore, it exceeded the performance of the existing agent-based methods. Although the first-round returns are relatively low in some cases, the damage caused by penalties is also relatively minor, and the fluctuation in long-term returns is significantly reduced. Adjusted Return increased by an average of 14.7 per cent. We hope that the Design Philosophy of FinGuard will provide some references for the construction of controllable and auditable agent systems for high-risk decision-making.

References

[1] Cao L. AI in finance: Challenges, techniques, and opportunities[J]. ACM Computing Surveys, 2023, 55(3): 1-38.

[2] Goodell J W, Kumar S, Lim W M, Pattnaik D. Artificial intelligence and machine learning in finance: Identifying foundations, themes, and research clusters from bibliometric analysis[J]. Journal of Behavioral and Experimental Finance, 2021, 32: 100577.

[3] Cohen G, Aiche A, Eichel R. Artificial intelligence models for predicting stock returns using fundamental, technical, and entropy-based strategies: A semantic-augmented hybrid approach[J]. Entropy, 2025, 27(6): 550.

[4] Mun Y, Kim N. Leveraging large language models for sentiment analysis and investment strategy development in financial markets[J]. Journal of Theoretical and Applied Electronic Commerce Research, 2025, 20(2): 77.

[5] Zhu P, Li Y, Liu Q, Cheng D, Jiang C. Financial time series prediction with multi-granularity graph augmented learning[J]. IEEE Transactions on Knowledge and Data Engineering, 2025, 37(11): 6436-6449.

[6] Botunac I, Petković T, Bosna J. Automated trading framework using LLM-driven features and deep reinforcement learning[J]. Big Data and Cognitive Computing, 2025, 9(12): 317.

[7] Lewis P, Perez E, Piktus A, Petroni F, Karpukhin V, Goyal N, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks[C]. Advances in Neural Information Processing Systems, 2020, 33: 9459-9474.

[8] Trivedi H, Balasubramanian N, Khot T, Sabharwal A. Interleaving retrieval with chain-of-thought reasoning for knowledge-intensive multi-step questions[C]. Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics, 2023: 10014-10037.

[9] Asai A, Wu Z, Wang Y, Sil A, Hajishirzi H. Self-RAG: Learning to retrieve, generate, and critique through self-reflection[C]. The Twelfth International Conference on Learning Representations, 2024.

[10] Luo S, Xu J, Zhang X, Wang L, Liu S, Hou H, Song L. RALLRec+: Retrieval augmented large language model recommendation with reasoning[J]. Expert Systems with Applications, 2026, 297: 129508.

[11] Liu Y, Fu L, Fu Y, Wu T, Bai H. Multi-hop reasoning over sparse knowledge graphs with deep reinforcement learning[J]. Expert Systems with Applications, 2025, 290: 128289.

[12] Jiang J, Zhou K, Dong Z, Ye K, Zhao X, Wen J R. StructGPT: A general framework for large language model to reason over structured data[C]. Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, 2023: 9237-9251.

[13] Madaan A, Tandon N, Gupta P, Hallinan S, Gao L, Wiegreffe S, et al. Self-Refine: Iterative refinement with self-feedback[C]. Advances in Neural Information Processing Systems, 2023, 36: 46534-46594.

[14] Shinn N, Cassano F, Berman E, Gopinath A, Narasimhan K, Yao S. Reflexion: Language agents with verbal reinforcement learning[C]. Advances in Neural Information Processing Systems, 2023, 36: 8634-8652.

[15] Yao S, Zhao J, Yu D, Du N, Shafran I, Narasimhan K R, Cao Y. ReAct: Synergizing reasoning and acting in language models[C]. The Eleventh International Conference on

Learning Representations, 2023.

[16] Hong S, Zhuge M, Chen J, Zheng X, Cheng Y, Wang J, et al. MetaGPT: Meta programming for a multi-agent collaborative framework[C]. The Twelfth International Conference on Learning Representations, 2024.

[17] Du Y, Li S, Torralba A, Tenenbaum J B, Mordatch I. Improving factuality and reasoning in language models through multiagent debate[C]. Proceedings of the 41st International Conference on Machine Learning, 2024.

[18] Wei J, Wang X, Schuurmans D, Bosma M, Ichter B, Xia F, et al. Chain-of-thought prompting elicits reasoning in large language models[C]. Advances in Neural Information Processing Systems, 2022, 35: 24824-24837.

[19] Chen W, Ma X, Wang X, Cohen W W. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks[J]. Transactions on Machine Learning Research, 2023.

[20] Gao L, Madaan A, Zhou S, Alon U, Liu P, Yang Y, et al. PAL: Program-aided language models[C]. Proceedings of the 40th International Conference on Machine Learning, 2023, 202: 10764-10799.

[21] Pan L, Albalak A, Wang X, Wang W. Logic-LM: Empowering large language models with symbolic solvers for faithful logical reasoning[C]. Findings of the Association for Computational Linguistics: EMNLP 2023, 2023: 3806-3824.

[22] Xi Z, Chen W, Guo X, He W, Ding Y, Hong B, et al. The rise and potential of large language model based agents: A survey[J]. Science China Information Sciences, 2025, 68(2): 121101.

[23] Bai X, Huang S, Wei C, Wang R. Collaboration between intelligent agents and large language models: A novel approach for enhancing code generation capability[J]. Expert Systems with Applications, 2025, 269: 126357.

[24] Huang J, Chen X, Mishra S, Zheng H S, Yu A W, Song X, Zhou D. Large language models cannot self-correct reasoning yet[C]. The Twelfth International Conference on Learning Representations, 2024.

[25] Gou Z, Shao Z, Gong Y, Shen Y, Yang Y, Duan N, Chen W. CRITIC: Large language models can self-correct with tool-interactive critiquing[C]. The Twelfth International Conference on Learning Representations, 2024.

[26] Shi F, Chen X, Misra K, Scales N, Dohan D, Chi E H, Schärli N, Zhou D. Large language models can be easily distracted by irrelevant context[C]. Proceedings of the 40th International Conference on Machine Learning, 2023, 202: 31210-31227.

[27] Brohi S, Mastoi Q u a, Jhanjhi N Z, Pillai T R. A research landscape of agentic AI and large language models: Applications, challenges and future directions[J]. Algorithms, 2025, 18(8): 499.

[28] Liu Y, Bu N, Li Z, Zhang Y, Zhao Z. AT-FinGPT: Financial risk prediction via an

audio-text large language model[J]. Finance Research Letters, 2025, 77: 106967.

[29] Dong Y, Wu F, Zhang K, Dai Y, Zhang S, Ye W, Chen S, Cheng Z Q. Large language model agents in finance: A survey bridging research, practice, and real-world deployment[C]. Findings of the Association for Computational Linguistics: EMNLP 2025, 2025: 17889-17907.

[30] James A, Trovati M, Bolton S. Retrieval-augmented generation to generate knowledge assets and creation of action drivers[J]. Applied Sciences, 2025, 15(11): 6247.

[31] Pan, L.; Albalak, A.; Wang, X.; Wang, W. Logic-LM: Empowering Large Language Models with Symbolic Solvers for Faithful Logical Reasoning. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2023, Singapore, 2023; pp. 3806-3824. https://doi.org/10.18653/v1/2023.findings-emnlp.248.

[32] Xi, Z.; Chen, W.; Guo, X.; He, W.; Ding, Y.; Hong, B.; Zhang, M.; Wang, J.; Jin, S.; Zhou, E.; et al. The rise and potential of large language model based agents: A survey. Science China Information Sciences 2025, 68, 121101. https://doi.org/10.1007/s11432-024-4222-0.

[33] Huang, J.; Chen, X.; Mishra, S.; Zheng, H.S.; Yu, A.W.; Song, X.; Zhou, D. Large language models cannot self-correct reasoning yet. arXiv preprint arXiv:2310.01798 2023.

[34] Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.R.; Cao, Y. React: Synergizing reasoning and acting in language models. In Proceedings of the The eleventh international conference on learning representations, 2022.

[35] Fırat, M.; Kuleli, S. What if GPT4 Became Autonomous: The Auto-GPT Project and Use Cases. Journal of Emerging Computer Technologies 2024, 3, 1-6. https://doi.org/10.57020/ject.1297961.

[36] Guo, Q.; Guo, Y.; Zhao, J. Unleashing the power of Large Language Models for Low-Resource Relation Triplet Extraction by Structure-to-Text Data Generation. Expert Systems with Applications 2026, 298, 129733. https://doi.org/10.1016/j.eswa.2025.129733.

[37] [37]Shinn, N.; Cassano, F.; Gopinath, A.; Narasimhan, K.; Yao, S. Reflexion: Language agents with verbal reinforcement learning. Advances in neural information processing systems 2023, 36, 8634-8652.

[38] Gou, Z.; Shao, Z.; Gong, Y.; Shen, Y.; Yang, Y.; Duan, N.; Chen, W. Critic: Large language models can self-correct with tool-interactive critiquing. arXiv preprint arXiv:2305.11738 2023.

[39] Brohi, S.; Mastoi, Q.u.a.; Jhanjhi, N.Z.; Pillai, T.R. A Research Landscape of Agentic AI and Large Language Models: Applications, Challenges and Future Directions. Algorithms 2025, 18. https://doi.org/10.3390/a18080499.

[40] Li, M., Shi, X., Leng, H., Zhou, W., Zheng, H.-T., & Zhang, K. (2023). Learning

Semantic Alignment with Global Modality Reconstruction for Video-Language Pre-training towards Retrieval. Proceedings of the AAAI Conference on Artificial Intelligence, 37(1), 1377–1385. https://doi.org/10.1609/aaai.v37i1.25222.

[41] L. Wang, M. Li and H. -T. Zheng, "Rethinking Rule-Based Approaches in Session-Based Recommendation," ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing(ICASSP), Rhodes Island, Greece, 2023, pp. 1-5, doi:10.1109/ICASSP49357.2023.10094851.

[42] M. Li, H. Liu, Y. Wang and H. -T. Zheng, "Bilingual Self-Attention Network: Generating Headlines for Online Linguistic Questions," 2021 International Joint Conference on Neural Networks (IJCNN), Shenzhen, China, 2021, pp. 1-8, doi: 10.1109/IJCNN52387.2021.9533288.

[43] Li, M., Wang, L., Liu, H., Wang, W., Zheng, HT. (2022). Context Reasoning Attention Network: Generating Plausible Distractors for Multi-choice Questions. In: Pimenidis, E., Angelov, P., Jayne, C., Papaleonidas, A., Aydin, M. (eds) Artificial Neural Networks and Machine Learning – ICANN 2022. ICANN 2022. Lecture Notes in Computer Science, vol 13531. Springer, Cham. https://doi.org/10.1007/978-3-031-15934-3_49