REASONING THROUGH VERIFIABLE FORECAST ACTIONS: CONSISTENCY-GROUNDED RL FOR FINANCIAL LLMS

Anonymous authors
Paper under double-blind review

ABSTRACT

Financial markets are characterized by extreme non-stationarity, low signal-to-noise ratios, and strong dependence on external information such as news, company fundamentals, and macroeconomic signals. Yet, existing approaches either abstract time-series into text or decouple forecasting from language-based reasoning, leading to a fundamental mismatch between qualitative reasoning and quantitative outcomes. To address this, we introduce **STOCK-R1**, a time-series–enhanced LLM that unifies stock forecasting and financial reasoning through a verifiable forecast action. Based on a tool-call design, the model first emits a forecast action, which is a structured and interpretable representation of its qualitative market outlook. It then invokes a time-series decoder conditioned on this action to generate distributional future trajectories, leading to more informed question answering and financial reasoning. We optimize the full pipeline with reinforcement learning, where rewards jointly reflect answer validity, forecast accuracy, and consistency between generated actions and observed time-series dynamics. In addition, rewards are reweighted by a sample-level uncertainty scalar, encouraging the model to accommodate varying uncertainty in market dynamics. We evaluate **STOCK-R1** on financial question answering and stock forecasting over a large-scale 10-year benchmark. **STOCK-R1** achieves strong performance across both tasks and improves question-answering accuracy over its corresponding direct-prompt Qwen3 backbones by 17.7 and 25.9 percentage points at the 4B and 8B scales, respectively. These findings demonstrate that structured forecast actions provide an effective interface between language reasoning and temporal prediction, enabling LLMs to reason through verifiable, interpretable, and numerically grounded decisions.

1 INTRODUCTION

Many high-stakes decision-making problems require reasoning over evolving systems whose behavior cannot be explained by a single modality. Financial markets are a particularly challenging instance: price movements are noisy, non-stationary, and tightly coupled with news, macroeconomic conditions and company fundamentals. Consequently, forecasting from past price trajectories alone remains difficult for both statistical models (Taylor & Letham, 2017; Ariyo et al., 2014; Box et al., 2015) and deep time-series architectures (Zeng et al., 2023; Liu et al., 2023; 2022; Nie, 2022), as predictive signals often extend beyond the observed numerical history.

To examine whether external context provides measurable predictive information, we conduct a simple non-parametric diagnostic: a k-nearest-neighbor regressor (Cunningham & Delany, 2021) predicts future cumulative returns by averaging outcomes from historically similar market states, with similarity computed from price history alone or together with textual contexts. Figure 1 shows that adding company-specific information, including business profiles, news, and fundamentals, consistently improves the Pearson correlation between predicted and

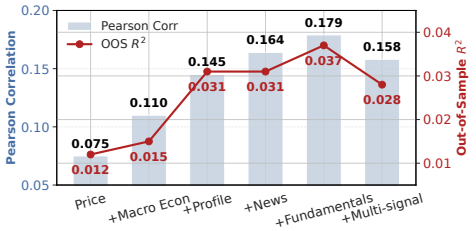


Figure 1: Pearson correlation and out-of-sample (OOS) R^2 for stock return prediction under different context settings.

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.

ground-truth returns and increases out-of-sample (OOS) R^2 from 0.012 to 0.037 (Campbell & Thompson, 2008). Although the absolute predictive signal remains modest, this suggests that textual market context contains measurable information beyond historical price dynamics.

These observations motivate multimodal financial modeling, but existing approaches lack a principled mechanism for coupling numerical forecasting with language-based reasoning. One line of work augments time-series models with news or fundamentals as auxiliary covariates (Li et al., 2024; Xu & Cohen, 2018; Li et al., 2023), improving predictive performance but offering limited capacity for open-ended reasoning. Another builds financial LLMs (Wu et al., 2023; Liu et al., 2025; Qian et al., 2025; Zhu et al., 2025; Xiao et al., 2025) that synthesize market narratives but remain limited in representing dense numerical sequences and performing precise quantitative reasoning (Wallace et al., 2019; Tan et al., 2024; Chen et al., 2025). Recent RL-based approaches improve temporal reasoning by training LLMs to generate reasoning traces or direct forecasts (Guan et al., 2025; Luo et al., 2025), yet still collapse semantic reasoning and numerical trajectory modeling into a single token-prediction objective, leaving their interaction weakly structured and difficult to verify. This suggests a natural question: *can we design an explicit, verifiable interface through which an LLM contributes contextual reasoning about market dynamics, while delegating trajectory-level numerical generation to a dedicated time-series decoder?*

This raises a central cross-modal alignment challenge: the LLM reasons over discrete semantic abstractions, whereas the decoder generates continuous trajectories. For effective synergy, the LLM must use contextual understanding to guide the decoder toward informed numerical generation; conversely, the generated trajectory should remain consistent with the context, reducing mismatches between reasoning and forecasting. This reflects a broader cross-modal reasoning challenge: aligning semantic reasoning with modality-specific generation without hallucinated outputs.

To address this gap, we introduce **STOCK-R1**, a financial LLM that couples language reasoning and numerical forecasting. Its core design is a *structured forecast action* produced by the policy LLM: a field-factorized profile of continuous, categorical, and ordinal attributes such as direction, drawdown, and turning-point behavior. This action serves as an executable signal for the time-series decoder, allowing the LLM's semantic assessment to guide numerical trajectory generation. It can also be evaluated against modality-derived evidence at the field level, providing a verifiable bridge between qualitative reasoning and quantitative forecasting.

Training proceeds in three stages. We first pretrain the time-series module to model future trajectory distributions and predictive uncertainty from historical market signals, then perform supervised fine-tuning (SFT) with modality-derived forecast profiles to warm up action generation and downstream reasoning. Finally, a GRPO-based (Shao et al., 2024) reinforcement learning stage optimizes the policy with a reward combining answer validity, forecast accuracy, and action-level consistency between structured fields and observed time-series behavior, where *uncertainty-aware reweighting* down-weights samples with high predictive uncertainty to improve robustness under noisy market regimes. Together with the structured action space, this design enables more precise credit assignment than unstructured free-form rationales. Empirically, **STOCK-R1** achieves strong performance in price movement forecasting and financial QA on a newly curated large-scale benchmark covering 10 years of S&P 500 stocks, suggesting that verifiable forecast actions provide an effective interface for coupling language reasoning with temporal prediction.

Our contributions are threefold: (1) We introduce forecast actions as an executable interface for effectively coupling language-based reasoning with trajectory-level time-series forecasting. (2) We propose a consistency-grounded reinforcement learning objective with uncertainty-aware reweighting, which aligns predictive commitments, generated trajectories, and final answers under noisy market dynamics. (3) We curate a large-scale financial benchmark with aligned temporal signals and textual context, together with an evaluation suite for forecasting, financial QA, and multimodal reasoning.

2 RELATED WORK

General Time-Series Forecasting. Time-series forecasting has evolved from statistical baselines (Ariyo et al., 2014; Taylor & Letham, 2018) to deep learning architectures, among which Transformer-based models have become dominant for capturing long-range dependencies (Zhou et al., 2021; Wu et al., 2021; Nie, 2022; Liu et al., 2023; Feng et al., 2024; Zhou et al., 2022; Kim

et al., 2025). Recent works adapt pre-trained LLMs to time series (Jin et al., 2023; Chang et al., 2023; Hu et al., 2025; Liu et al., 2024a), typically by aligning time-series embeddings with the text space of frozen LLMs, and this line has progressed toward time-series foundation models (Rasul et al., 2023; Goswami et al., 2024; Shi et al., 2024; Liu et al., 2024b; Woo et al., 2024; Ansari et al., 2024) that emphasize large-scale pretraining across diverse datasets, generalization, and transferability across tasks and domains. However, these methods mainly target relatively stationary forecasting benchmarks and rarely integrate auxiliary multimodal context to explain their predictions.

Stock Prediction and Quantitative Modeling. Financial forecasting is challenging due to stochastic market dynamics. Previous approaches (Ariyo et al., 2014; Feng et al., 2019; Shi et al., 2025; Lee et al., 2025; Duan et al., 2022) extract patterns from historical stock data, implicitly assuming that past prices contain sufficient information for future prediction, which contradicts the Efficient Market Hypothesis (Malkiel, 1989). Recent multimodal approaches (Kim et al., 2019; Xu & Cohen, 2018; Li et al., 2024; Wang et al., 2024) augment time-series models with external information such as textual news or relational graphs to capture cross-asset and event-driven dependencies, but typically treat auxiliary modalities as feature-level enhancements for prediction. In contrast, our method natively handles reasoning-centric financial tasks, enabling coherent forecasting and answer generation.

Financial LLMs and Trading Agents. Financial LLMs (Wu et al., 2023; Wang et al., 2023; Liu et al., 2025; Qian et al., 2025; Zhu et al., 2025) have attracted growing interest; while excelling in knowledge-intensive tasks, they struggle with forward-looking trend prediction. In parallel, agent-based and RL-driven frameworks (Sun et al., 2026; Xiao et al., 2024; 2025) focus on portfolio or risk optimization and interactive market play, but do not natively support fine-grained forecasting or reasoning about future markets in context-rich QA settings. Our work explores an orthogonal and complementary direction, emphasizing grounded forecasting to support complex financial QA.

3 METHODOLOGY

As shown in Figure 2, **STOCK-R1** operates as an action-conditioned tool-use pipeline for financial reasoning. Given multimodal stock inputs (Sec. 3.1), we first pre-train a multichannel time-series encoder that generates future trajectories and uncertainty estimates (Sec. 3.2). The policy LLM then emits a structured forecast action that invokes the time-series decoder (Sec. 3.3), whose output trajectory is returned as numerical evidence for the answer. The pipeline is initialized with joint SFT (Sec. 3.4) and optimized through consistency-grounded RL with uncertainty-aware updates (Sec. 3.5).

3.1 MULTIMODAL TIME-SERIES DATA FORMULATION

The fundamental challenge in financial time-series forecasting is market non-stationarity: statistical patterns shift over time, making those learned from past observations unreliable under future conditions and weakening generalization. To mitigate this, we reframe the pre-training objective from deterministic point forecasting to probabilistic density estimation (Lo & MacKinlay, 1988). Given a multichannel input $\mathbf{X}_t = [O_{t-L:t}, H_{t-L:t}, L_{t-L:t}, C_{t-L:t}, V_{t-L:t}] \in \mathbb{R}^{5\times L}$ of Open, High, Low, Close, and Volume (OHLCV) over a lookback window L, we transform them into log returns, $r_t^{\mathrm{N}} = \ln(\frac{\mathrm{N}_t}{C_{t-1}})$ for $\mathrm{N} \in \{O, H, L, C\}$ and $r_t^V = \ln(\frac{V_t+1}{V_{t-1}+1})$, which converts the non-stationary sequence into a quasi-stationary signal suitable for representation learning (Fama, 1970).

To complement these numerical signals with semantic context, we align them with a structured modality $\mathcal{T}_t$, including: (1) **Macroeconomic Data**: Key indicators such as US Treasury yields (*e.g.,* 2Y, 10Y spreads) and inflation metrics that drive market sentiment; (2) **Company Fundamentals**: Quarterly snapshots including revenue, operating margins, and leverage ratios (*e.g.,* debt-to-equity); and (3) **Business & News Context**: A summary of the company's core operations alongside recent headlines and sentiment analysis. These heterogeneous sources are serialized into a unified textual prompt that grounds the model's reasoning (see Appendix A for collection details and examples).

3.2 MULTI-SIGNAL TIME SERIES ENCODER PRE-TRAINING

Time Series Encoder Architecture. We employ a causal attention-based transformer that uniformly encodes the multichannel input $\mathbf{X}_t \in \mathbb{R}^{5\times L}$ to learn joint representations across temporal and variable dimensions. To preserve temporal consistency in this flattened space, we apply Rotary Positional

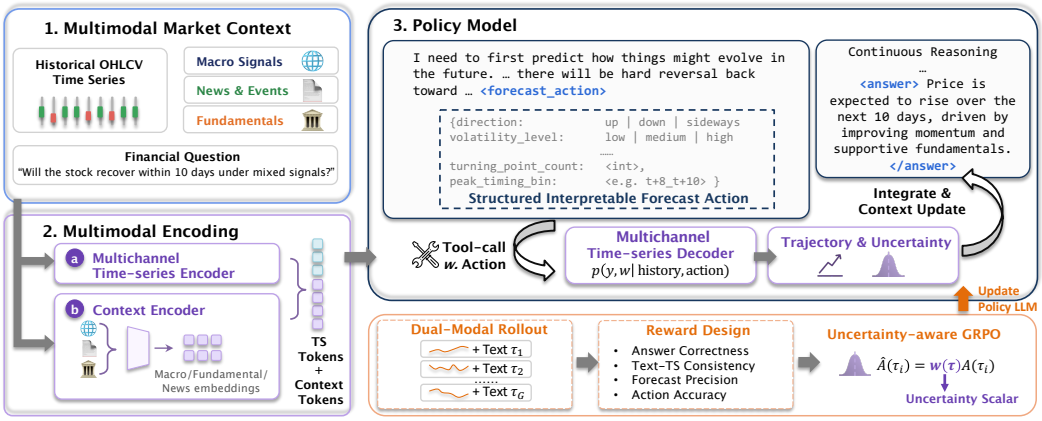


Figure 2: Overview of **STOCK-R1**. The model encodes multimodal market context into unified tokens, then uses a policy LLM to generate a structured forecast action that conditions a multichannel time-series decoder. The predicted trajectory and uncertainty are returned to the LLM for grounded financial reasoning, with the whole policy optimized via uncertainty-aware dual-modal rollout.

Embeddings (RoPE) (Su et al., 2024) with positional indices assigned by timestamp t. Our Multivariate Time Attention injects a learnable binary bias $\mathbf{B}$ into the attention scores to differentiate intra-variable dependencies (variable indices $v_i = v_j$) from inter-variable cross-correlations ($v_i \neq v_j$), complemented by a causal block-diagonal mask $\mathbf{M}$ ensuring predictions at time t depend solely on the history $\tau \leq t$ across all channels. More details are given in Appendix B.1.

Pretraining Objectives. We train the time-series encoder with a reconstructive objective to produce flattened latent tokens $\mathbf{H}_{\text{ts}} \in \mathbb{R}^{K \times d_m}$, where K is the number of time-series tokens and d_m is the LLM's model dimension. These tokens are prepended to the textual tokens in the LLM input, allowing the policy model to jointly attend to numerical market dynamics and semantic market context. Since point-wise forecasting ignores the inherent stochasticity of market dynamics, we project $\mathbf{H}_{\text{ts}}$ to model the conditional probability of future returns as dynamic Gaussian distributions $\mathcal{N}(\mu_{t+s}, \sigma^2_{t+s})$ for Open, Close, and Volume. For the intraday extremes (High and Low), which reflect the volatility boundaries, we employ quantile regression to estimate tail risks. The training objective $\mathcal{L}_{\text{TS}}$ combines the Negative Log-Likelihood (NLL) and the quantile loss ρ_τ over the horizon S:

$$\mathcal{L}_{\text{TS}} = \sum_{s=1}^{S} \sum_{i \in \{O,C,V\}} \lambda_i \left(\frac{(r^i_{t+s} - \mu^i_{t+s})^2}{2(\sigma^i_{t+s})^2} + \frac{1}{2} \ln\left((\sigma^i_{t+s})^2\right) \right) + \sum_{j \in \{H,L\}} \lambda_j \rho_{\tau_j} \left(r^j_{t+s} - q^{(j,\tau_j)}_{t+s} \right),$$

where the quantile loss $\rho_\tau(u) = u(\tau - \mathbb{I}_{u<0})$ targets the 95th percentile for the High price and the 5th percentile for the Low (*i.e.,* $\tau_H = 0.95; \tau_L = 0.05$). Beyond a predictive mean, this formulation explicitly quantifies market uncertainty as the aggregate variance $\mathcal{U} = \sum_{s,i} (\sigma^i_{t+s})^2$, which is further normalized to a bounded range to serve as a calibrated stability signal for downstream RL.

3.3 STRUCTURED FORECAST ACTIONS

Single-Action Tool-Call Interface. The pretrained time-series model generates future trajectories from historical signals but does not capture how market context should shape the forecast. We therefore formulate forecasting-enhanced reasoning as a single-action tool-call process, where the LLM emits a structured forecast action (via `<forecast_action>`) as the argument to a dedicated time-series decoder. This action serves as a semantic-to-numeric interface, converting the LLM's market assessment into an executable control signal that modulates the decoder's forecast distribution.

Action Space Definition. Let the forecast action consist of K fields, each describing one temporal attribute of the future trajectory, such as trend direction, end-point movement, turning-point behavior, or peak timing. The domain $\mathcal{A}_k$ of the k-th field may be continuous, categorical, or ordinal: *e.g.,* numerical summaries such as end-point change take continuous values, a categorical field such as trend direction has $\mathcal{A}_k = \{up, flat, down\}$, and an ordinal field such as volatility level uses discretized bins $\mathcal{A}_k = \{1, \ldots, B_k\}$. The complete action space is the Cartesian product $\mathcal{A}_{\text{forecast}} = \mathcal{A}_1 \times \cdots \times \mathcal{A}_K$.

Given the multimodal context s and user query q, the policy LLM π_θ generates a forecast action $a = (a_1, \ldots, a_K) \in \mathcal{A}_{\text{forecast}}$ by $a \sim \pi_\theta(\cdot \mid s, q)$. This field-factorized construction makes the action interpretable, exposes fine-grained temporal commitments, and enables field-level supervision and reward computation that are difficult to obtain from free-form rationales.

Action-conditioned Forecasting Tool. Given the encoded historical embedding $\mathbf{H}_{\text{ts}}$ and a structured action a, we normalize its continuous fields, embed its categorical and ordinal fields, and concatenate them into a unified representation, which an MLP action encoder $e_\psi(\cdot)$ maps to the continuous conditioning vector $\mathbf{z}_a$. The time-series decoder $\phi(\cdot)$ then generates future trajectories conditioned on both the historical representation and encoded action, $\hat{y}_{t+1:t+S} \sim p_\phi(\cdot \mid \mathbf{H}_{\text{ts}}, e_\psi(a))$. The decoder output is serialized as the observation returned by `<forecast_action>`, and the LLM completes the final answer conditioned on the original context, structured action, and numerical trajectory.

Why Forecast Actions. Forecast actions bridge the LLM's semantic reasoning and the decoder's numerical generation while preserving cross-modal faithfulness. This intermediate action space renders the interaction (i) **controllable**, as distinct actions induce distinct predictive distributions over future trajectories under the same historical context; (ii) **verifiable**, as each field of the action can be evaluated against both the subsequently generated forecast and the realized outcome; and (iii) **trainable**, as RL can assign credit to each attribute rather than solely to the entire prediction.

3.4 Joint SFT with Context-Aware Forecasting Decoder

To bootstrap supervised training, we distill cold-start trajectories from GPT-5 under the same tool-call format. Given a multimodal market context, the teacher generates financial questions covering diverse perspectives, constructs action-grounded reasoning, and completes question-specific reasoning and answers. During this outcome-conditioned data construction, the target action is deterministically computed from the realized future trajectory rather than generated by the teacher, and guides the construction and verification of the reasoning trace. The realized future and its derived action are used only for supervision; at inference time, the policy receives causally available inputs and must emit the structured action before invoking `<forecast_action>`. More details are given in Appendix A.3.

In this stage, the policy LLM is trained to generate a forecast rationale, emit a valid and precise structured forecast action a, consume the decoder trajectory returned by `<forecast_action>`, and produce the final reasoning trace and answer. The overall objective $\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{SFT}} + \lambda_{\text{ts}}\mathcal{L}_{\text{TS}}$ contains two coupled terms with separate gradient paths: $\mathcal{L}_{\text{TS}}$ (Sec. 3.2) trains the time-series module with the distribution-aware loss, conditioned on the target forecast action, while $\mathcal{L}_{\text{SFT}}$ trains the LLM with next-token prediction to follow the reasoning–action–tool–answer format. The two modules are coupled through the same target action and realized trajectory.

3.5 Reinforcement Learning for Grounded Reasoning

After SFT, the model can follow the tool-call format, but supervised learning alone does not guarantee that the generated action, decoder trajectory, and final answer remain mutually consistent. We therefore introduce an uncertainty-aware reinforcement learning (RL) stage over the full dual-modal rollout: for each state $s = (\mathbf{X}_t, \mathcal{T}_t, q)$, the policy samples a textual forecast action $\hat{a}$, the time-series decoder generates a numerical trajectory $\hat{y}_{t+1:t+S} \sim p_\phi(\cdot \mid \mathbf{H}_{\text{ts}}, e_\psi(\hat{a}))$, and the LLM produces an answer $\hat{z}$ conditioned on the original context, action, and trajectory. During RL, the pretrained time-series module, including the uncertainty estimator and action-conditioned decoder, remains fixed; only the policy LLM is updated. This rollout explicitly couples the LLM's reasoning with the action-conditioned decoder: a poor action can degrade the forecast, and an inconsistent forecast can reduce the reward assigned to the reasoning trace. Since financial labels are intrinsically noisy, the RL update also accounts for the predictive uncertainty $\mathcal{U}_q$ from Sec. 3.2.

Reward Design. The reward evaluates both modalities:

$$R = R_{\text{ans}} + \alpha R_{\text{act}} + \beta R_{\text{prec}} + \gamma R_{\text{cons}},$$

where α, β, and γ are weighting coefficients and each component targets a distinct failure mode:

- **Answer Correctness** (R_{ans}) evaluates the final answer by task type: exact matching over the target class for categorical or decision-oriented queries, and tolerance-aware scoring for open numerical questions.

- **Action Accuracy** (R_{act}) compares the generated action $\hat{a}$ field by field with the ground-truth action $a^\star = g(y_{t+1:t+S})$ derived from the realized future trajectory: categorical fields are scored by exact matching, ordinal fields receive distance-aware partial credit, and continuous summary fields use scale-normalized error.
- **Forecast Precision** (R_{prec}) measures the accuracy of the decoder-generated trajectory against the realized time series, using scale-normalized errors for price and volume and direction-aware metrics for return movement to keep the reward comparable across assets.
- **Action-Trajectory Consistency** (R_{cons}) verifies whether the structured action is consistent with the decoder-generated trajectory: we parse the trajectory into a profile $\tilde{a}_h = g(\hat{y}_{t+1:t+S})$ and compare it with the policy action $\hat{a}_h$ field by field. Categorical fields are scored by exact agreement, ordinal fields receive distance-sensitive similarity, and continuous summaries are compared using scale-normalized error.

The consistency term is central to our alignment mechanism. Conditioning alone does not guarantee that the fixed decoder uses every action field, as it may still rely mainly on the historical representation $\mathbf{H}_{\text{ts}}$. It therefore penalizes, for example, cases where the policy emits a bullish action while the decoder returns a bearish trajectory, or claims low volatility while producing a wide forecast range. This internal constraint does not by itself establish correctness with respect to the realized future, which is separately enforced by R_{act} and R_{prec}; together, these terms assign credit to policy actions that are both executable by the fixed forecasting tool and supported by observed outcomes.

Uncertainty-Aware Consistency-Grounded GRPO. For each query, we sample a group of dual-modal rollouts $\{o_i\}_{i=1}^{G}$, each containing the generated action, numerical trajectory, and final answer. The group-relative advantage is computed from the total reward above, allowing the model to compare alternative actions and answers under the same market state. To avoid overfitting noisy credit assignment in unstable market regimes, we further reweight the GRPO-style objective (Shao et al., 2024) with the normalized forecast uncertainty $\mathcal{U}_q$ predicted by the fixed time-series encoder, which is not produced by the policy being optimized and is treated as a stop-gradient sample weight:

$$\mathcal{J}_{\text{u-GRPO}}(\theta) = \mathbb{E}_{q,\{o_i\}}\boxed{w(\mathcal{U}_q)}\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\left\{\min\left[r_t(\theta)\hat{A}_{i,t}, \text{clip}(r_t(\theta))\hat{A}_{i,t}\right] - \beta\mathbb{D}_{KL}[\pi_\theta||\pi_{ref}]\right\}.$$

We normalize the sample weight $w(\mathcal{U}_q)$ using a threshold-based exponential decay (Appendix B.4), which reduces, but does not remove, the contribution of samples with high predictive uncertainty, thereby limiting the influence of noisy supervision while retaining nonzero learning signals from uncertain market regimes. Predictive uncertainty is not assumed to be equivalent to difficulty, realized volatility, or event rarity; rather, it estimates the reliability of outcome-based credit assignment.

4 Experiments

We evaluate **Stock-R1** on complex financial QA (Sec. 4.1) and time-series forecasting (Sec. 4.2), followed by ablation studies and analysis of forecast-action quality (Sec. 4.3). Appendices C.4 and C.5 further provide a scalability analysis and an investment simulation of buy/sell signals.

4.1 Reasoning-based Financial QA

Experimental Setting. We evaluate predictive and financial question answering across five task categories designed to assess both numerical foresight and the ability to integrate heterogeneous financial signals. We compare **Stock-R1**, instantiated on Qwen3-4B and Qwen3-8B, against strong language-model baselines, including general-purpose LLMs (Llama3 (Dubey et al., 2024), Qwen3 (Yang et al., 2025), GPT-5, OpenAI-o3) and domain-adapted financial reasoning models (Fin-o1 (Qian et al., 2025), Fin-R1 (Liu et al., 2025), DianJin-R1 (Zhu et al., 2025)). All models are evaluated using task-specific accuracy, with configurations provided in Appendix C.2.

Results. Table 1 shows that **Stock-R1**-8B achieves the highest average accuracy among the evaluated models, while **Stock-R1**-4B outperforms the same-scale baselines and financial LLMs, consistent with the benefit of coupling language reasoning with temporal generation. The most pronounced gain appears in Forecast QA, where general-purpose LLMs perform poorly, suggesting that textual reasoning alone provides limited support for precise forward-looking numerical prediction.

STOCK-R1 substantially improves this category by grounding its reasoning in trajectory-level time-series forecasts. Beyond pure forecasting, **STOCK-R1** remains competitive across Event Detection, Fundamental QA, News Analysis, and Multi-signal Reasoning, indicating that the action-conditioned forecasting interface can also support broader context-aware financial reasoning. Overall, these results suggest that verifiable forecast actions provide a practical mechanism for aligning market-context interpretation with numerical forecasting evidence. Additional controls (chain-of-thought, few-shot prompting, and an unconditioned time-series tool) and QA case studies are reported in Appendices D.2 and C.6.

Table 1: Performance comparison across financial QA tasks. Best performance is in **bold**, while the second-best is underlined.

Model	Forecast QA	Event Detection	Fundamental QA	News Analysis	Multi-signal Reasoning	Avg.
General-purpose LLM						
Llama3-3B	2.53	48.87	45.68	42.72	56.54	39.27
Llama3-8B	3.21	44.10	47.13	51.77	54.81	40.20
Qwen3-4B	7.92	65.58	48.94	35.11	57.82	43.07
Qwen3-8B	6.23	39.42	47.06	44.79	52.69	38.04
Qwen3-30B	10.34	55.35	45.40	51.41	63.39	45.18
Llama3-70B	5.35	53.25	59.34	55.28	60.85	46.81
DeepSeek-V3.2	3.37	56.90	44.58	42.00	60.95	41.56
GPT-5	8.16	51.08	52.69	60.49	61.24	46.73
OpenAI-o3	2.78	53.43	57.76	57.89	67.07	47.79
Financial LLM						
Fin-R1-8B	40.28	49.35	56.41	60.48	60.28	53.36
Fin-o1-14B	32.86	31.89	41.27	53.53	**69.12**	45.73
DianJin-R1-32B	38.42	63.71	52.47	62.40	62.58	55.92
STOCK-R1-4B	51.00	**68.52**	59.70	61.83	63.02	60.81
STOCK-R1-8B	**56.78**	67.49	**61.08**	**67.16**	67.45	**63.99**

4.2 TIME SERIES FORECASTING

Table 2: Holistic Forecasting Performance Comparison. We evaluate **STOCK-R1** against Zero-shot Foundation Models, Supervised Time Series Models, and Multimodal LLMs. **(U)** denotes Unimodal input (Time Series only), and **(M)** denotes Multimodal input (Time Series + Text). The best performance is in **bold**, while the second-best performance is underlined.

	Model	Price ↓ (MAPE)	Price ↓ (rMSE)	Volume ↓ (MAPE)	Volume ↓ (rMSE)	Return ↓ (rMSE)	3-day ↑ Acc. (%)	5-day ↑ Acc. (%)	10-day ↑ Acc. (%)	Volatility ↓ (MAE×10^{-3})
Time Series Foundation Models (**Zero-shot**)	Timer-XL	0.946	1.460	6.758	1.392	0.955	44.7	43.8	40.3	7.212
	Moirai	0.031	1.395	7.293	1.367	0.048	53.2	52.8	54.5	2.709
	MOMENT	0.076	1.484	6.150	1.409	0.104	49.2	48.5	46.6	2.937
	Chronos-T5	0.031	1.411	7.097	1.376	0.051	50.5	50.1	49.6	6.453
	Time-MoE	0.035	1.437	5.759	**1.358**	0.052	48.5	48.6	48.4	3.008
Time Series Models (**Train-from-scratch**)	ARIMA	0.035	1.405	6.134	1.415	0.061	53.4	53.0	50.0	–
	Prophet	0.058	1.416	6.128	1.377	0.083	47.3	45.9	49.9	2.966
	DLinear	0.038	1.433	6.019	1.388	0.055	51.8	51.7	50.8	3.015
	Mamba	0.037	1.448	6.345	1.464	0.053	52.3	53.0	53.3	3.146
	TimeMixer	0.031	1.401	6.087	1.403	0.049	53.8	54.6	55.2	2.946
	TimesNet	0.031	1.423	6.037	1.387	0.046	52.2	50.3	53.5	2.898
	PatchTST	0.037	1.409	5.991	1.395	0.055	52.9	51.1	51.7	14.124
	NSTransformer	0.031	1.391	6.074	1.399	0.048	53.3	51.1	53.2	2.928
	iTransformer	0.032	1.415	6.098	1.399	0.046	54.4	53.8	53.3	2.967
	AutoTimes	0.042	1.444	5.938	1.384	0.059	52.9	51.4	49.4	2.634
	Time-LLM	0.049	1.450	6.108	1.411	0.070	50.2	51.4	46.9	3.383
	Kronos	0.036	1.371	5.725	1.385	0.051	50.8	52.5	54.2	2.634
	FSCA	**0.029**	1.365	5.983	1.394	0.042	54.9	53.2	53.5	2.835
	STOCK-R1 (U)	**0.029**	1.339	**5.316**	1.384	0.041	55.1	55.3	55.0	**2.537**
Language Models (**Multimodal Input**)	Qwen3-4B	0.036	1.320	7.945	1.422	0.050	56.4	55.6	58.6	2.778
	Qwen3-8B	0.044	**1.303**	8.373	1.417	0.060	55.6	55.9	59.2	2.682
	Llama3-3B	0.039	1.328	–	–	0.057	50.5	52.3	53.1	2.884
	Llama3-8B	0.035	1.324	5.393	1.418	0.049	55.1	55.7	57.8	2.758
	GPT-5	0.032	1.334	8.409	1.411	0.046	55.2	54.6	56.6	2.903
	STOCK-R1 (M)	0.031	1.319	5.872	1.380	**0.040**	**57.6**	**58.8**	**59.6**	2.602

Baselines. We evaluate **STOCK-R1** on time-series forecasting against three categories of baselines: (1) Time Series Foundation Models evaluated zero-shot (*e.g.,* Moirai (Woo et al., 2024), Chronos (Ansari et al., 2024)); (2) Time Series Models trained from scratch, including general forecasters (ARIMA and Prophet (Box et al., 2015; Taylor & Letham, 2017)), the re-pretrained stock-specific model Kronos (Shi et al., 2025), and state-of-the-art baselines such as AutoTimes (Liu et al., 2024a), Time-LLM (Jin et al., 2023), and FSCA (Hu et al., 2025), which leverage LLMs to incorporate textual summaries but do not naturally support lengthy, temporally aligned multimodal context, often limiting their ability to model high-frequency market dynamics; and (3) Multimodal Large Language Models (*e.g.,* GPT-5, Qwen3 (Yang et al., 2025)) that understand time series in text

space. Unless a pretrained model requires its native input interface, forecasting baselines use the same log-return and log-volume transformations as Sec. 3.1, with a 90-day lookback and a 10-day prediction horizon. We report our model in Unimodal **(U)** and Multimodal **(M)** settings to quantify the gain from textual grounding. Performance is measured using Mean Absolute Percentage Error (MAPE) and Root Mean Square Error (rMSE) for numerical precision (Price, Volume, Returns), Directional Accuracy (Acc.) over the next 3, 5, and 10 days, and Mean Absolute Error (MAE) for volatility estimation. More experimental details are given in Appendix C.1.

Results Analysis. As shown in Table 2, among pure time-series approaches, **STOCK-R1 (U)** achieves state-of-the-art or highly competitive performance across numerical precision, return prediction, and volatility estimation. Notably, it obtains the best volatility MAE, suggesting that the distribution-aware pretraining objective better captures market risk than point-forecasting losses. Comparing **(U)** and **(M)**, multimodal context brings the clearest gains in directional accuracy across 3-, 5-, and 10-day horizons while numerical precision remains broadly comparable, consistent with the role of textual signals in financial markets: news, fundamentals, and macro context often provide directional cues but do not necessarily determine exact price levels. Overall, **STOCK-R1 (M)** outperforms general-purpose LLMs by combining semantic market context with a dedicated forecasting module, yielding stronger directional prediction without sacrificing trajectory-level numerical fidelity.

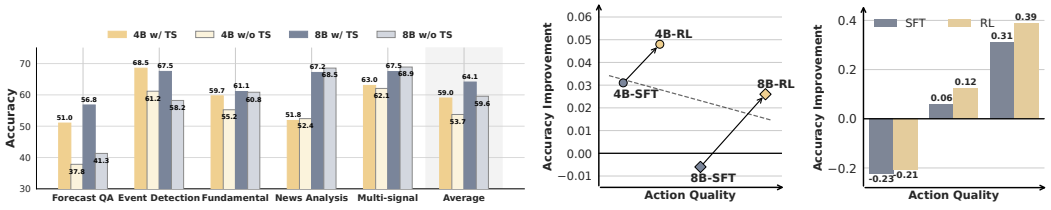
4.3 ABLATION STUDIES AND ANALYSIS

Action Conditioning Beyond Tool Access. To test whether the structured action contributes beyond simply providing the LLM with a numerical forecast, Table 3 compares an unconditioned tool, $p_\phi(y \mid \mathbf{H}_{\text{ts}})$, with the proposed action-conditioned tool, $p_\phi(y \mid \mathbf{H}_{\text{ts}}, e_\psi(a))$. Action conditioning improves 10-day directional accuracy by 4.1 and average QA accuracy by 3.81 percentage points. The substantially higher trajectory–action alignment further indicates that the structured action functions as an effective control signal rather than merely accompanying an otherwise unchanged numerical forecast.

Table 3: Tool-interface ablation at the 4B scale.

Metric	Unconditioned	Action-cond.
Return rMSE ↓	0.043	**0.040**
10-day Acc. ↑	55.5	**59.6**
Avg. QA ↑	57.0	**60.8**
Traj.–Action ↑	12.7	**53.8**

Time-Series Tokens Improve Forecasting Performance. Figure 3a examines the impact of explicitly injecting time series as soft tokens across financial reasoning tasks. Time-series tokens yield consistent gains at both 4B and 8B scales, indicating that the benefits are robust to model capacity. The largest relative improvements appear in Forecast QA, where explicit temporal dynamics are critical for anticipating future movements and models without time-series tokens struggle to infer quantitative trends from textual context alone. For other question types, the gains are more moderate but still consistent. Overall, the improved average accuracy across all tasks and scales suggests that time-series tokens provide a useful inductive bias for both predictive and reasoning-intensive financial QA.



(a) Performance comparison across QA tasks *w/* and *w/o* time-series token.

(b) RL improves action quality beyond SFT

(c) Better actions lead to higher accuracy

Figure 3: Ablation and Analysis of Time-Series Grounding and Action Quality

RL Enhances Action Quality and Improves Directional Accuracy. We measure action quality by field-level accuracy, comparing each generated forecast-action field with the corresponding target field derived from the realized future trajectory. As shown in Figure 3b, RL primarily refines the model's forecast action space, systematically shifting the model toward higher-quality and more internally consistent actions, suggesting that its core effect is to improve the precision and reliability of the intermediate decision interface. Moreover, Figure 3c shows a strong correlation between action quality and directional accuracy improvement, with Pearson correlation coefficients exceeding

0.65 across the evaluated settings. Better-aligned forecast actions are thus associated with more accurate directional predictions, and the RL stage further strengthens this relationship by enforcing qualitative–numerical consistency during training. This monotonic trend supports the structured action as an effective bottleneck between high-level reasoning and numerical prediction.

Uncertainty-Aware RL Stabilizes Cross-Modal Alignment. Table 4 shows that u-GRPO achieves the best average accuracy, while removing either the action reward R_{act} or the forecast precision reward R_{prec} leads to performance degradation. The larger drop from removing R_{act} highlights the importance of explicitly supervising the intermediate forecast action. Figure 4 further shows that uncertainty-aware reweighting stabilizes RL training: without uncertainty weighting, policy entropy drops more sharply in the early stage, indicating premature convergence and reduced exploration. In contrast, uncertainty-aware reweighting maintains higher entropy by down-weighting high-uncertainty, low-reliability samples, which helps preserve exploration and yields stronger late-stage performance. These results suggest that the reward design improves both cross-modal credit assignment and optimization robustness under noisy market dynamics.

Consistency Reward Improves Numerical Grounding. To probe how RL helps beyond task accuracy, we use LLM-based rubrics to evaluate numerical grounding across four dimensions, including answer–evidence consistency, trajectory–action alignment, logical validity, and action success rate (detailed in Appendix C.7). As shown in Figure 5, removing RL entirely causes the largest degradation, confirming that reinforcement learning is essential not only for answer correctness but also for aligning the model's qualitative outlook, numerical forecast, and final reasoning. More revealingly, removing only the consistency reward R_{cons} preserves answer–evidence consistency but disproportionately harms trajectory–action alignment, indicating that final-answer supervision alone cannot ensure the structured forecast action remains numerically compatible with the generated trajectory. The consistency reward thus provides fine-grained credit assignment at the action–trajectory interface, preventing forecast actions that are semantically plausible but numerically unsupported.

Table 4: Ablation study of RL variants and reward components.

Method	Avg. QA Acc.
SFT *w/* thinking	57.3 ± 0.8
+ GRPO	66.0 ± 1.1
+ u-GRPO	$\mathbf{67.5 \pm 0.6}$
w/o R_{act} (Actions)	63.2 ± 0.8
w/o R_{prec} (Forecast)	66.4 ± 1.1

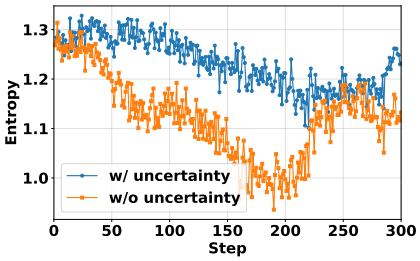


Figure 4: The entropy curve with and without uncertainty-aware reweighting.

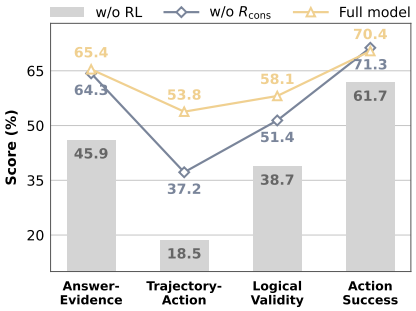


Figure 5: Ablation of RL components on numerical grounding metrics

5 Conclusion

We introduce **Stock-R1**, a time-series-enhanced financial LLM that couples language-based market reasoning with trajectory-level numerical forecasting through executable forecast actions. Instead of requiring the LLM to directly generate dense future trajectories, **Stock-R1** uses forecast actions as an intermediate interface to guide a dedicated time-series decoder and to support verifiable alignment between reasoning, forecasting, and final answers. Consistency-grounded RL with uncertainty-aware reweighting further improves credit assignment under noisy, non-stationary market dynamics. On our historical benchmark, **Stock-R1** achieves strong performance in both financial QA and time-series forecasting, while the matched tool-interface comparison shows additional gains from action conditioning beyond access to an unconditioned numerical forecast. These findings support structured forecast actions as an effective interface for grounding language-model reasoning in modality-specific temporal prediction. Because the current evaluation is retrospective and focuses on S&P 500 equities, prospective and broader cross-domain validation remain important next steps.

AI USE STATEMENT

Generative AI tools were used in three roles in this work. First, GPT-5 was used as part of the research methodology to synthesize questions and cold-start reasoning trajectories under the tool-call format described in Appendix A.3; the resulting samples were subjected to tool-aware verification and filtering before use. Second, LLM-based rubrics were used as evaluation instruments for numerical grounding and reasoning consistency, as documented in Appendix C.7. Third, generative AI tools assisted with language polishing, clarity, and organization of the manuscript. The authors reviewed the AI-assisted data, evaluations, and text and take responsibility for the final content, claims, and artifacts in this work.

ETHICS STATEMENT

This study does not involve human subjects or interventions. It uses historical market data, company fundamentals, macroeconomic indicators, and news from the sources and collection process described in Appendix A. Use and release of third-party data and content should comply with the providers' terms and applicable laws. Financial forecasting systems may produce inaccurate, biased, or stale outputs and could cause financial harm if treated as investment advice or deployed without oversight. The reported results are intended for research rather than automated trading or financial advice.

REPRODUCIBILITY STATEMENT

The model architecture, forecast-action interface, training objectives, and evaluation protocol are described in Sections 3.1–3.5 and Section 4. Appendix A documents data collection, temporal alignment, preprocessing, dataset statistics, and cold-start construction; Appendices B.1–B.4 give additional method details; and Appendices C.1–D.4 report experimental settings, hyperparameters, evaluation rubrics, and additional controls.

REFERENCES

Abdul Fatir Ansari, Lorenzo Stella, Caner Turkmen, Xiyuan Zhang, Pedro Mercado, Huibin Shen, Oleksandr Shchur, Syama Sundar Rangapuram, Sebastian Pineda Arango, Shubham Kapoor, et al. Chronos: Learning the language of time series. *arXiv preprint arXiv:2403.07815*, 2024.

Adebiyi A Ariyo, Adewumi O Adewumi, and Charles K Ayo. Stock price prediction using the arima model. In *2014 UKSim-AMSS 16th international conference on computer modelling and simulation*, pp. 106–112. IEEE, 2014.

George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

John Y Campbell and Samuel B Thompson. Predicting excess stock returns out of sample: Can anything beat the historical average? *The Review of Financial Studies*, 21(4):1509–1531, 2008.

Ching Chang, Wen-Chih Peng, and Tien-Fu Chen. Llm4ts: Two-stage fine-tuning for time-series forecasting with pre-trained llms. *CoRR*, 2023.

Jialin Chen, Aosong Feng, Ziyu Zhao, Juan Garza, Gaukhar Nurbek, Cheng Qin, Ali Maatouk, Leandros Tassiulas, Yifeng Gao, and Rex Ying. Mtbench: A multimodal time series benchmark for temporal reasoning and question answering. *arXiv preprint arXiv:2503.16858*, 2025.

Padraig Cunningham and Sarah Jane Delany. K-nearest neighbour classifiers-a tutorial. *ACM computing surveys (CSUR)*, 54(6):1–25, 2021.

Yitong Duan, Lei Wang, Qizhong Zhang, and Jian Li. Factorvae: A probabilistic dynamic factor model based on variational autoencoder for predicting cross-sectional stock returns. In *Proceedings of the AAAI conference on artificial intelligence*, volume 36, pp. 4468–4476, 2022.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd of models. *arXiv e-prints*, pp. arXiv–2407, 2024.

Eugene F Fama. Efficient capital markets: A review of theory and empirical work. *The journal of Finance*, 25(2):383–417, 1970.

Aosong Feng, Jialin Chen, Juan Garza, Brooklyn Berry, Francisco Salazar, Yifeng Gao, Rex Ying, and Leandros Tassiulas. Efficient high-resolution time series classification via attention kronecker decomposition. *arXiv preprint arXiv:2403.04882*, 2024.

Fuli Feng, Xiangnan He, Xiang Wang, Cheng Luo, Yiqun Liu, and Tat-Seng Chua. Temporal relational ranking for stock prediction. *ACM Transactions on Information Systems (TOIS)*, 37(2): 1–30, 2019.

Mononito Goswami, Konrad Szafer, Arjun Choudhry, Yifu Cai, Shuo Li, and Artur Dubrawski. Moment: A family of open time-series foundation models. *arXiv preprint arXiv:2402.03885*, 2024.

Tong Guan, Zijie Meng, Dianqi Li, Shiyu Wang, Chao-Han Huck Yang, Qingsong Wen, Zuozhu Liu, Sabato Marco Siniscalchi, Ming Jin, and Shirui Pan. Timeomni-1: Incentivizing complex reasoning with time series in large language models. *arXiv preprint arXiv:2509.24803*, 2025.

Yuxiao Hu, Qian Li, Dongxiao Zhang, Jinyue Yan, and Yuntian Chen. Context-alignment: Activating and enhancing llm capabilities in time series. *arXiv preprint arXiv:2501.03747*, 2025.

Ming Jin, Shiyu Wang, Lintao Ma, Zhixuan Chu, James Y Zhang, Xiaoming Shi, Pin-Yu Chen, Yuxuan Liang, Yuan-Fang Li, Shirui Pan, et al. Time-llm: Time series forecasting by reprogramming large language models. *arXiv preprint arXiv:2310.01728*, 2023.

Jongseon Kim, Hyungjoon Kim, HyunGi Kim, Dongjun Lee, and Sungroh Yoon. A comprehensive survey of deep learning for time series forecasting: architectural diversity and open challenges. *Artificial Intelligence Review*, 58(7):1–95, 2025.

Raehyun Kim, Chan Ho So, Minbyul Jeong, Sanghoon Lee, Jinkyu Kim, and Jaewoo Kang. Hats: A hierarchical graph attention network for stock movement prediction. *arXiv preprint arXiv:1908.07999*, 2019.

Hyunwoo Lee, Jihyeong Jeon, Jaemin Hong, and U Kang. Mitigating distribution shift in stock price data via return-volatility normalization for accurate prediction. In *Proceedings of the 34th ACM International Conference on Information and Knowledge Management*, pp. 1458–1467, 2025.

Shuqi Li, Weiheng Liao, Yuhan Chen, and Rui Yan. Pen: prediction-explanation network to forecast stock price movement with better explainability. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pp. 5187–5194, 2023.

Shuqi Li, Yuebo Sun, Yuxin Lin, Xin Gao, Shuo Shang, and Rui Yan. Causalstock: Deep end-to-end causal discovery for news-driven multi-stock movement prediction. *Advances in Neural Information Processing Systems*, 37:47432–47454, 2024.

Yong Liu, Haixu Wu, Jianmin Wang, and Mingsheng Long. Non-stationary transformers: Exploring the stationarity in time series forecasting. *Advances in neural information processing systems*, 35: 9881–9893, 2022.

Yong Liu, Tengge Hu, Haoran Zhang, Haixu Wu, Shiyu Wang, Lintao Ma, and Mingsheng Long. itransformer: Inverted transformers are effective for time series forecasting. *arXiv preprint arXiv:2310.06625*, 2023.

Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Autotimes: Autoregressive time series forecasters via large language models. *Advances in Neural Information Processing Systems*, 37:122154–122184, 2024a.

Yong Liu, Guo Qin, Xiangdong Huang, Jianmin Wang, and Mingsheng Long. Timer-xl: Long-context transformers for unified time series forecasting. *arXiv preprint arXiv:2410.04803*, 2024b.

Zhaowei Liu, Xin Guo, Fangqi Lou, Lingfeng Zeng, Jinyi Niu, Zixuan Wang, Jiajie Xu, Weige Cai, Ziwei Yang, Xueqian Zhao, et al. Fin-r1: A large language model for financial reasoning through reinforcement learning. *arXiv preprint arXiv:2503.16252*, 2025.

Andrew W Lo and A Craig MacKinlay. Stock market prices do not follow random walks: Evidence from a simple specification test. *The review of financial studies*, 1(1):41–66, 1988.

Yucong Luo, Yitong Zhou, Mingyue Cheng, Jiahao Wang, Daoyu Wang, Tingyue Pan, and Jintao Zhang. Time series forecasting as reasoning: A slow-thinking approach with reinforced llms. *arXiv preprint arXiv:2506.10630*, 2025.

Burton G Malkiel. Efficient market hypothesis. In *Finance*, pp. 127–134. Springer, 1989.

Y Nie. A time series is worth 64words: Long-term forecasting with transformers. *arXiv preprint arXiv:2211.14730*, 2022.

Lingfei Qian, Weipeng Zhou, Yan Wang, Xueqing Peng, Han Yi, Yilun Zhao, Jimin Huang, Qianqian Xie, and Jian-yun Nie. Fino1: On the transferability of reasoning-enhanced llms and reinforcement learning to finance. *arXiv preprint arXiv:2502.08127*, 2025.

Kashif Rasul, Arjun Ashok, Andrew Robert Williams, Arian Khorasani, George Adamopoulos, Rishika Bhagwatkar, Marin Biloš, Hena Ghonia, Nadhir Hassen, Anderson Schneider, et al. Lag-llama: Towards foundation models for time series forecasting. In *R0-FoMo: Robustness of Few-shot and Zero-shot Learning in Large Foundation Models*, 2023.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Xiaoming Shi, Shiyu Wang, Yuqi Nie, Dianqi Li, Zhou Ye, Qingsong Wen, and Ming Jin. Time-moe: Billion-scale time series foundation models with mixture of experts. *arXiv preprint arXiv:2409.16040*, 2024.

Yu Shi, Zongliang Fu, Shuo Chen, Bohan Zhao, Wei Xu, Changshui Zhang, and Jian Li. Kronos: A foundation model for the language of financial markets. *arXiv preprint arXiv:2508.02739*, 2025.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Rui Sun, Yifan Sun, Sheng Xu, Li Zhao, Jing Li, Daxin Jiang, Chen Hua, and Zuo Bai. Trade-r1: Bridging verifiable rewards to stochastic environments via process-level reasoning verification. *arXiv preprint arXiv:2601.03948*, 2026.

Mingtian Tan, Mike A Merrill, Vinayak Gupta, Tim Althoff, and Thomas Hartvigsen. Are language models actually useful for time series forecasting? *Advances in Neural Information Processing Systems*, 37:60162–60191, 2024.

Sean J Taylor and Benjamin Letham. Prophet: forecasting at scale. *PeerJ Preprints*, 5:e3190v2, 2017.

Sean J Taylor and Benjamin Letham. Forecasting at scale. *The American Statistician*, 72(1):37–45, 2018.

Eric Wallace, Yizhong Wang, Sujian Li, Sameer Singh, and Matt Gardner. Do nlp models know numbers? probing numeracy in embeddings. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pp. 5307–5315, 2019.

Neng Wang, Hongyang Yang, and Christina Dan Wang. Fingpt: Instruction tuning benchmark for open-source large language models in financial datasets. *arXiv preprint arXiv:2310.04793*, 2023.

Shengkun Wang, Taoran Ji, Linhan Wang, Yanshen Sun, Shang-Ching Liu, Amit Kumar, and Chang-Tien Lu. Stocktime: A time series specialized large language model architecture for stock price prediction. *arXiv preprint arXiv:2409.08281*, 2024.

Gerald Woo, Chenghao Liu, Akshat Kumar, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Unified training of universal time series forecasting transformers. 2024.

Haixu Wu, Jiehui Xu, Jianmin Wang, and Mingsheng Long. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Advances in neural information processing systems*, 34:22419–22430, 2021.

Shijie Wu, Ozan Irsoy, Steven Lu, Vadim Dabravolski, Mark Dredze, Sebastian Gehrmann, Prabhanjan Kambadur, David Rosenberg, and Gideon Mann. Bloomberggpt: A large language model for finance. *arXiv preprint arXiv:2303.17564*, 2023.

Yijia Xiao, Edward Sun, Di Luo, and Wei Wang. Tradingagents: Multi-agents llm financial trading framework. *arXiv preprint arXiv:2412.20138*, 2024.

Yijia Xiao, Edward Sun, Tong Chen, Fang Wu, Di Luo, and Wei Wang. Trading-r1: Financial trading with llm reasoning via reinforcement learning. *arXiv preprint arXiv:2509.11420*, 2025.

Yumo Xu and Shay B Cohen. Stock movement prediction from tweets and historical prices. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1970–1979, 2018.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Ailing Zeng, Muxi Chen, Lei Zhang, and Qiang Xu. Are transformers effective for time series forecasting? In *Proceedings of the AAAI conference on artificial intelligence*, volume 37, pp. 11121–11128, 2023.

Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI conference on artificial intelligence*, volume 35, pp. 11106–11115, 2021.

Tian Zhou, Ziqing Ma, Qingsong Wen, Xue Wang, Liang Sun, and Rong Jin. Fedformer: Frequency enhanced decomposed transformer for long-term series forecasting. *arXiv preprint arXiv:2201.12740*, 2022.

Jie Zhu, Qian Chen, Huaixia Dou, Junhui Li, Lifan Guo, Feng Chen, and Chi Zhang. Dianjin-r1: Evaluating and enhancing financial reasoning in large language models. *arXiv preprint arXiv:2504.15716*, 2025.

A Dataset Curation

A.1 Data Collection and Preprocessing

To construct a robust multimodal benchmark, we aggregate high-frequency market data and unstructured textual signals from distinct providers, ensuring strict temporal synchronization.

Numerical Data Acquisition and Preprocessing.. Historical daily pricing data, covering Open, High, Low, Close, and Volume (OHLCV) for the S&P 500 universe, is retrieved via the `yfinance` API[1]. To mitigate the inherent non-stationarity of raw prices and facilitate efficient representation learning, we apply a log-return transformation to the raw inputs $\mathbf{X}_t$. Specifically, raw price components are normalized relative to the previous day's closing price, formally defined as $r_t^{\mathrm{N}} = \ln(\frac{\mathrm{N}_t}{C_{t-1}})$ for $\mathrm{N} \in \{O, H, L, C\}$. Similarly, the volume sequence is log-normalized as $r_t^V = \ln(\frac{V_t+1}{V_{t-1}+1})$ to reduce the impact of outliers.

Textual Data Collection and Alignment.. Semantic context is sourced from public market API[2] to complement the numerical signals. We enforce a rigorous alignment protocol to prevent look-ahead bias and ensure causal validity. For any given trading day t:

1. **Company Fundamentals**: We retrieve the most recently published quarterly financial statement prior to t (*e.g.,* revenue, operating margins, leverage ratios).
2. **Business & News Context**: We aggregate news articles from a fixed lookback window using strict keyword matching based on the company's ticker and official name, retaining only information available before market close at $t-1$.
3. **Macroeconomic Data**: Key indicators such as US Treasury yields (*e.g.,* 2Y vs. 10Y spreads) and inflation metrics are similarly aligned by selecting the latest available release prior to time t.

These heterogeneous data sources are cleaned and serialized into a unified structured prompt $\mathcal{T}_t$, serving as the grounding context for the model.

A.2 Dataset Statistics

Time-Series Data. Our numerical dataset consists of daily equity time series spanning from **2015-01-01 to 2025-10-01** over the **S&P 500** universe. For each stock, we record five standard market channels, *i.e., Open, High, Low, Close,* and *Volume* (OHLCV), at daily resolution. After aligning trading calendars and removing missing or inactive tickers, the dataset contains approximately 2,700 trading days per asset. All price-related channels are transformed into log-returns relative to the previous close to mitigate non-stationarity, while volume is log-normalized. The resulting multivariate time series serves as the numerical backbone for both forecasting and reasoning tasks.

Question Answering Data. On top of the aligned time-series and textual context, we construct a large-scale financial QA dataset covering five task categories: *Forecast QA, Event Detection, Fundamental QA, News Analysis,* and *Multi-signal Reasoning*. The dataset contains 40,000 training questions and 4,000 evaluation questions, each paired with temporally aligned numerical trajectories, macroeconomic indicators, company fundamentals, and the 10 most recent news contexts. All questions are grounded at a specific decision time t, with answers constrained to use only information available prior to or at t to avoid look-ahead bias. Each QA instance includes a structured reasoning trace and a final answer, enabling both supervised fine-tuning and reinforcement learning for time-series-grounded reasoning.

A.3 Cold-start Data Generation

Synthesizing high-quality supervision via model distillation is essential for bootstrapping reasoning in specialized financial domains, where generalist models often struggle to jointly integrate numerical dynamics and semantic context (Liu et al., 2025; Wang et al., 2023; Shao et al., 2024). Based on the unified multimodal context, we construct cold-start trajectories that follow a tool-call interface. Concretely, GPT-5 first produces a concise market analysis, then emits a structured action followed by `<forecast_action>` that specifies forecast-relevant fields. This action is used to condition

[1] `https://ranaroussi.github.io/yfinance/`
[2] `https://massive.com/`

the time-series decoder, and the returned trajectory is provided to the model for final reasoning and a concise answer in <answer>. Our pipeline consists of three stages, each targeting a specific source of noise or inefficiency in synthetic supervision:

1. **Domain-Aware Question Synthesis**. We prompt GPT-5 to generate questions across diverse financial tasks (*e.g., Forecast QA*, *Event Detection*, *News Analysis*, and *Multi-signal Reasoning*) to cover varied reasoning horizons and decision objectives.
2. **Action-grounded Reasoning Construction**. For each query, we derive a target forecast action from the realized future trajectory and require the teacher trace to justify an action that is consistent with available historical, textual, and macroeconomic evidence. This produces supervision for the reasoning before the intermediate action that will consequently control the decoder.
3. **Tool-aware Verification and Filtering**. We execute the action-conditioned <forecast_action> to obtain a numerical trajectory, verify whether the trajectory supports the target answer, and filter out samples that are either inconsistent or trivially solvable by a smaller baseline (*e.g.,* Qwen3-4B). This yields $\mathcal{D}_{\text{SFT}}$ with explicit reasoning-action-answer alignment.

A.4 EXAMPLES

Examples of Questions

Forecasting QA Questions:
- What will be the maximum drawdown within the next x trading days?
- What will be the percentage change in the closing price over the next x days??

Price-action Event Detection Questions:
- Will the stock drop more than 5% from today's close within the next x days?
- Will the stock hit more than 5% from today's close in the next x days?

Fundamental Questions:
- Given the current 10-year vs 2-year Treasury yield spread, is the stock predicted to have higher volatility or lower volatility in the next 10 days?
- If the 1-month yield exceeds the 10-year yield (yield curve inversion), will the stock's next-10-day cumulative return stay positive?

News Analysis Questions:
- Given the 10 most recent headlines, will the stock's predicted 7-day cumulative return stay positive?
- If most news sentiment is negative, will the stock still recover above today's price within 10 days?

Multi-signal Reasoning Questions:
- Given the current market conditions, will the stock's predicted volatility rise or fall?
- Combining macro shifts, news flow, and price forecasts, will you recommend a scale-in, scale-out, or stay-flat strategy?

Examples of Multi-signal Input

Ticker: XYZ
Date: 2025-06-20
Company: Block, Inc. (XYZ)
Business Summary: Founded in 2009, Block provides payment services to merchants, along with related services. The company also launched Cash App, a person-to-person payment network. In 2024, Square's payment volume was almost USD 250 million.
Macro Snapshot:
- US 1M Treasury yield: 4.20%
- US 3M Treasury yield: 4.39%
- US 2Y Treasury yield: 3.90%
- US 10Y Treasury yield: 4.38%
- US 30Y Treasury yield: 4.89%

Fundamentals (Q1 2025):
- revenue: 5.77B

- gross profit: 2.29B
- operating income: 329.30M
- net income: 188.72M
- gross margin: 40.00%
- operating margin: 6.00%
- net margin: 3.00%
- operating cash flow margin: 2.00%
- total assets: 36.40B
- total liabilities: 14.96B
- shareholders' equity: 21.44B
- current ratio: 2.27
- quick ratio: 2.26
- liabilities-to-equity: 0.70
- debt-to-assets: 0.41
- working capital: 11.03B
- asset turnover: 0.16
- effective tax rate: 17.00%
- cash from operations: 133.34M
- cash from investing: 914.71M
- cash from financing: -1.21B
- net cash flow: -141.34M

News:

- **2025-06-08: The 3 Best Growth Stocks to Buy With $100 Right Now**
  The article discusses three growth stocks that have been beaten down by the market but could produce incredible returns going forward: Marvell Technology, Block, and DraftKings. It highlights the growth prospects and potential value of these companies.
  Sentiment: positive
  Sentiment Reason: The article believes Block's Cash App and Square businesses have long-term growth potential, and the recent sell-off in the stock price presents a good opportunity for growth investors.
- **2025-05-31: Better Growth Stock: Block vs. SoFi Technologies**
  The article compares two fintech companies, SoFi Technologies and Block (formerly Square), and suggests that SoFi has a more promising future than Block. SoFi is benefiting from the growing trend of online banking, while Block's growth is slowing due to increased competition in the payment processing space and uncertainty around the adoption of blockchain technology.
  Sentiment: negative
  Sentiment Reason: The article suggests that Block's growth is slowing due to increased competition in the payment processing space and uncertainty around the adoption of blockchain technology, which the company has prioritized.
  ...

B METHOD DETAILS

B.1 TIME-SERIES ENCODER

Additional Details on Multi-Signal Time-Series Encoder. We provide a more explicit formulation of the multi-signal time-series encoder used in Sec. 3.2. Given a multichannel input sequence $\mathbf{X}_t \in \mathbb{R}^{C\times L}$ with channel index $v \in \{1,\dots,C\}$ and temporal index $\tau \le t$, we flatten the sequence into a token stream while preserving temporal order. Each token embedding is augmented with Rotary Positional Embeddings (RoPE), where the rotation angle is a deterministic function of the absolute timestamp τ, ensuring that relative temporal offsets are encoded directly in the query–key inner product. Formally, for attention head h, the rotated queries and keys are given by $\tilde{\mathbf{q}}_\tau^{(h)} = \mathrm{RoPE}(\mathbf{q}_\tau^{(h)}, \tau), \quad \tilde{\mathbf{k}}_\tau^{(h)} = \mathrm{RoPE}(\mathbf{k}_\tau^{(h)}, \tau).$

To enforce causal consistency across all variables, we introduce a block-diagonal causal mask $M \in \{0, -\infty\}^{(CL)\times(CL)}$, where

$$M_{(\tau,v),(\tau',v')} = \begin{cases} 0, & \tau' \le \tau, \\ -\infty, & \text{otherwise.} \end{cases}$$

This mask guarantees that representations at time τ only attend to historical observations from all channels. In addition, we inject a learnable binary bias matrix B into the attention logits to distinguish intra-variable dependencies from inter-variable correlations:

$$B_{(\tau,v),(\tau',v')} = \begin{cases} b_{\text{intra}}, & v = v', \\ b_{\text{inter}}, & v \neq v', \end{cases}$$

where b_{intra} and b_{inter} are trainable scalars shared across layers. The resulting attention computation for head h is

$$\text{Attn}^{(h)}(\tau, v) = \sum_{\tau',v'} \text{softmax}\left(\frac{\tilde{\mathbf{q}}^{(h)}_{\tau,v} \cdot \tilde{\mathbf{k}}^{(h)}_{\tau',v'}}{\sqrt{d_h}} + M_{(\tau,v),(\tau',v')} + B_{(\tau,v),(\tau',v')} \right) \mathbf{v}^{(h)}_{\tau',v'}.$$

The encoder outputs a sequence of latent time-series tokens $\mathbf{H}_{\text{ts}} \in \mathbb{R}^{K\times d_m}$, which are subsequently projected to parametric distribution heads. For each forecasting horizon $s \in \{1, \ldots, S\}$ and target variable $i \in \{O, C, V\}$, we obtain the mean and variance parameters via linear projections:

$$\mu^{(i)}_{t+s} = \mathbf{w}^{(i)\top}_{\mu} \mathbf{h}^{\text{ts}}_t, \quad \sigma^{(i)}_{t+s} = \text{softplus}\left(\mathbf{w}^{(i)\top}_{\sigma} \mathbf{h}^{\text{ts}}_t \right),$$

where $\mathbf{h}^{\text{ts}}_t$ denotes the aggregated latent state corresponding to time t. The softplus activation ensures positivity of σ and stabilizes training. For high and low prices, separate heads predict quantile estimates $q^{(H,\tau_H)}_{t+s}$ and $q^{(L,\tau_L)}_{t+s}$, enabling asymmetric modeling of upside and downside risk. Together, this design allows the encoder to produce both point forecasts and calibrated uncertainty estimates, which are later consumed by the forecasting decoder and the uncertainty-aware reinforcement learning stage.

B.2 Structured Forecast Actions

```
{
  "future_window": "t+1_t+10",
  "start_value": <float>,
  "end_value": <float>,
  "max_value": <float>,
  "min_value": <float>,
  "mean_close": <float>,

  "direction": "<up | down | flat>",
  "end_change_pct": <float>,
  "range_pct": <float>,
  "volatility_level": "<low | medium | high>",
  "range_width_bin": "<narrow | moderate | wide>",
  "max_drawdown_bin": "<low | medium | high>",

  "turning_point_count": <int>,
  "peak_timing_bin": "<early | middle | late>",
  "trough_timing_bin": "<early | middle | late>",
  "monotonicity": "<increasing | decreasing | mixed>",
  "trendline_fit": "<weak | moderate | strong>",
  "tail_risk_level": "<low | medium | high>"
}
```

Each action is implemented as a JSON-style dictionary containing three groups of fields: numerical summary statistics of the future trajectory, categorical or ordinal market states, and temporal shape

descriptors. During supervised data construction, the target action is deterministically derived from the realized future trajectory, providing field-level supervision for the LLM's intermediate forecast decision. During inference, the LLM emits the same structured action before invoking `<forecast_action>`; the action fields are embedded and passed through an MLP action encoder to produce the conditioning vector for the time-series decoder. This design makes the LLM's market outlook explicit, executable, and field-wise verifiable.

B.3 Reinforcement Learning Preliminary

Our RL training is based on GRPO algorithm (Shao et al., 2024), which is a variant of policy-gradient optimization designed for large language models, where per-query baselines are estimated from a group of sampled rollouts rather than a learned value function.

Rollout Generation. Given an input query q, the current policy π_θ generates a group of G independent rollouts $\{o_i\}_{i=1}^{G}$, where each rollout $o_i = (a_{i,1}, \ldots, a_{i,T_i})$ corresponds to a complete response trajectory (*e.g.,* reasoning steps and final answer). Each rollout is evaluated by a scalar reward function $R_i = R(o_i)$, which may incorporate correctness, formatting, or task-specific criteria.

Group-Relative Advantage. Instead of estimating a value baseline, GRPO computes a group-relative advantage by normalizing rewards within each rollout group: $\hat{A}_i = \frac{R_i - \text{Mean}(\mathbf{R})}{\text{Std.}(\mathbf{R})}$. This formulation encourages rollouts that perform better than the group average while suppressing uniformly poor generations, providing a low-variance baseline without additional critic training.

Optimization Objective. GRPO adopts a PPO-style clipped objective applied at the token level. For each rollout o_i, the objective is defined as

$$\mathcal{L}_{\text{GRPO}}(\theta) = \mathbb{E}_{q,\{o_i\}}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\min\Big(r_{i,t}(\theta)\hat{A}_i,\ \text{clip}(r_{i,t}(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_i\Big) - \beta\,\text{KL}(\pi_\theta \,\|\, \pi_{\text{ref}})\right], \tag{1}$$

where $r_{i,t}(\theta) = \frac{\pi_\theta(a_{i,t}|q,a_{i,<t})}{\pi_{\text{ref}}(a_{i,t}|q,a_{i,<t})}$, is the likelihood ratio between the current policy and a frozen reference policy π_{ref}. The KL penalty regularizes policy updates and stabilizes training, while the clipping parameter ϵ prevents excessively large gradient steps. In this work, GRPO provides the foundation upon which we introduce uncertainty-aware reweighting to better handle stochastic and noisy financial environments (Sec. 3.5).

B.4 Uncertainty Reweighting

Uncertainty Weighting Formulation. To modulate the learning signal based on market volatility, we calculate the sample weight $w(\mathcal{U}_q)$ using an exponential decay function anchored to a stability threshold u_0. The formulation is defined as:

$$w(\mathcal{U}_q) = \begin{cases} 1.0 & \text{if } \mathcal{U}_q \le u_0 \\ exp\left(-\alpha \cdot \frac{\mathcal{U}_q - u_0}{u_{\text{hi}} - u_0}\right) & \text{if } \mathcal{U}_q > u_0 \end{cases} \tag{2}$$

where α is a hyperparameter controlling the decay rate, and $u$hi represents the upper bound used for normalization.

C Experiments

C.1 Experimental Settings

To rigorously evaluate the performance of our proposed method, we benchmark against state-of-the-art time series forecasting models, including training-from-scratch ones and foundation models.

Evaluation Metrics for Time-Series Forecasting. For the n-th instance, let the historical close at the last observed day be $c_n^{(0)}$ and the future close trajectory over horizon $H = 10$ be $\{c_{n,h}\}_{h=1}^{H}$, with

predictions $\{\hat{c}_{n,h}\}_{h=1}^{H}$. We report (i) **Price** errors: mean absolute percentage error (MAPE) and root mean squared error (RMSE).

$$\mathrm{MAPE}_{\text{price}} = \frac{1}{NH}\sum_{n=1}^{N}\sum_{h=1}^{H}\left|\frac{\hat{c}_{n,h}-c_{n,h}}{c_{n,h}}\right|, \quad \mathrm{RMSE}_{\text{price}} = \sqrt{\frac{1}{NH}\sum_{n=1}^{N}\sum_{h=1}^{H}\big(z(\hat{c}_{n,h})-z(c_{n,h})\big)^2}, \tag{3}$$

where $z(\cdot)$ is instance-wise normalization (applied per sample across the horizon to reduce scale effects).

(ii) **Return** metrics based on simple returns relative to the historical close (with $\epsilon > 0$ for numerical stability):

$$r_{n,h} = \frac{c_{n,h}}{\max(c_n^{(0)},\epsilon)} - 1, \qquad \hat{r}_{n,h} = \frac{\hat{c}_{n,h}}{\max(c_n^{(0)},\epsilon)} - 1, \tag{4}$$

and daily-return RMSE across all horizons,

$$\mathrm{RMSE}_{\text{return}} = \sqrt{\frac{1}{NH}\sum_{n=1}^{N}\sum_{h=1}^{H}\big(\hat{r}_{n,h}-r_{n,h}\big)^2}. \tag{5}$$

Directional accuracy at horizon $k \in \{3,5,10\}$ uses cumulative return $\hat{R}_n^{(k)} = \hat{r}_{n,k}$ and $R_n^{(k)} = r_{n,k}$:

$$\mathrm{DA}_k = \frac{1}{N}\sum_{n=1}^{N}\mathbb{I}\Big[\mathrm{sign}(\hat{R}_n^{(k)}) = \mathrm{sign}(R_n^{(k)})\Big]. \tag{6}$$

(iii) **Volatility** is evaluated by realized volatility over the predicted/true close paths as the sum of squared log-returns:

$$\begin{aligned}\mathrm{RV}_n &= \sum_{h=1}^{H-1}\Big(\log\max(c_{n,h+1},\epsilon) - \log\max(c_{n,h},\epsilon)\Big)^2,\\ \widehat{\mathrm{RV}}_n &= \sum_{h=1}^{H-1}\Big(\log\max(\hat{c}_{n,h+1},\epsilon) - \log\max(\hat{c}_{n,h},\epsilon)\Big)^2.\end{aligned} \tag{7}$$

We report $\mathrm{MAE}_{\text{RV}} = \frac{1}{N}\sum_{n=1}^{N}\big|\widehat{\mathrm{RV}}_n - \mathrm{RV}_n\big|$ and the coefficient of determination.

(iv) **Volume** metrics follow the same horizon-based aggregation on the future volume series $\{v_{n,h}\}$ and predictions $\{\hat{v}_{n,h}\}$, using instance-wise normalization $z(\cdot)$:

$$\begin{aligned}\mathrm{MAPE}_{\text{vol}} &= \frac{1}{NH}\sum_{n=1}^{N}\sum_{h=1}^{H}\left|\frac{z(\hat{v}_{n,h})-z(v_{n,h})}{\max(z(v_{n,h}),\epsilon)}\right|,\\ \mathrm{RMSE}_{\text{vol}} &= \sqrt{\frac{1}{NH}\sum_{n=1}^{N}\sum_{h=1}^{H}\big(z(\hat{v}_{n,h})-z(v_{n,h})\big)^2}.\end{aligned} \tag{8}$$

C.2 Hyper-parameter Settings

For time-series encoder pretraining, we use a 90-day input window to predict the next 10 trading days. The encoder adopts a patch length of 5 with dropout set to 0.1. We sweep model capacities by varying the hidden dimension $d_{\text{model}} \in \{512, 768, 1024\}$, paired with $\{8, 12, 16\}$ attention heads and $\{10, 16\}$ Transformer layers. All models are trained for 50 epochs using AdamW with learning rate 1×10^{-4} and weight decay 0.01, a cosine learning-rate schedule with 1000 warmup steps. Unless otherwise stated, pretraining uses market data spanning 2015-01-01 to 2025-01-01 and test from 2025-01-01 to 2025-10-01. Table 5 summarizes the architectural configurations of the time-series encoders at different scales.

For supervised fine-tuning (SFT), we fine-tune `Qwen3-4B-Instruct-2507` and `Qwen3-8B` using LoRA with rank $r = 32$, scaling factor $\alpha = 32$, applied to projection and feed-forward modules,

Table 5: Architecture configurations of the time-series encoder at different scales.

Encoder Size	Hidden Dim	# Layers	# Attention Heads	# Parameters
Small	512	8	8	27M
Base	768	12	12	65M
Large	1024	16	16	119M

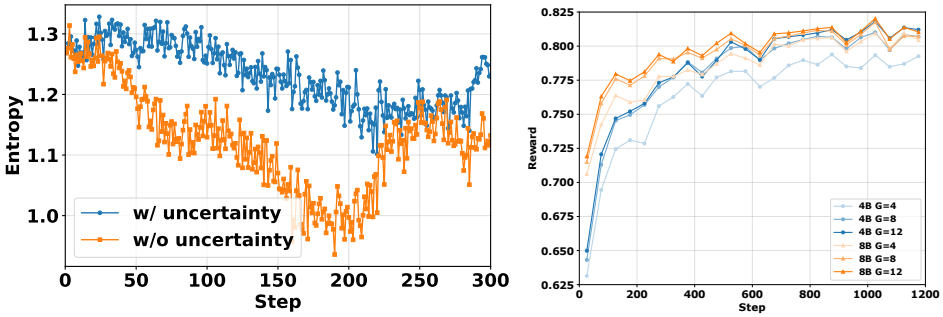


Figure 6: **Left:** Policy entropy over training steps, comparing training with and without uncertainty-aware reweighting. **Right:** Reward convergence under different group sizes.

and initialize the model with pre-trained time-series encoder checkpoints. Supervised fine-tuning is conducted with the AdamW optimizer with learning rate 2×10^{-5} and weight decay 0.01, cosine warmup of 100 steps for 5 epochs.

For reinforcement learning, we employ VERL with uncertainty-aware reweighting of the learning signal. We select the reward weights through validation search and use $\alpha = 1.0$, $\beta = 0.5$, and $\gamma = 1.0$ as the default setting, which empirically yields the best trade-off among action accuracy, forecast precision, and action–trajectory consistency. The uncertainty parameters are set to $u_0 = 0.25$, $u_{\text{hi}} = 0.50$, $w_{\text{hi}} = 0.50$, and $u_{\text{cap}} = 1.0$ in Eq. 2. Group size G is tuned over $\{4, 8, 12\}$ and the default configuration is set to $G = 8$.

C.3 Effect of Uncertainty and Rollout Size

Figure 6 (Left) illustrates the role of uncertainty-aware reweighting in stabilizing training dynamics and improving downstream performance. As shown in the left panel, training without uncertainty leads to a faster and more pronounced decrease in policy entropy in the early stage of RL training, indicating premature loss of exploration and increasingly deterministic behavior. This entropy collapse is also accompanied by higher instability across training steps. Instead, uncertainty-aware reweighting maintains consistently higher entropy, encouraging sustained exploration by down-weighting high-volatility and low-reliability samples. This difference in learning dynamics translates into better late-stage performance: while the model trained without uncertainty saturates early, the uncertainty-aware reweighting continues to improve and achieves higher accuracy ultimately. These results suggest that uncertainty-aware reweighting mitigates premature convergence, preserves exploration under noisy market regimes, and leads to more robust policy optimization.

Figure 6 (Right) studies the effect of the group size G on training reward convergence for both **Stock-R1**-4B and **Stock-R1**-8B. Across both model scales, we observe that increasing G generally improves reward stability and final convergence, as larger groups provide a more reliable estimate of relative advantages. The impact is more pronounced for the smaller 4B model, indicating higher variance in policy updates and suboptimal optimization. In contrast, the 8B model is more robust to smaller group sizes, though it still benefits from moderate increases in G. To balance optimization quality and training efficiency, we therefore select $G = 8$ as the default rollout number.

C.4 Scalability Analysis

We investigate the scalability of the proposed time-series encoder along two critical dimensions: pre-training data size and model capacity. As illustrated in Figure 7, we systematically vary the training data duration by shifting the start date from 2015 to 2024, while keeping the test set fixed. Simultaneously, we scale the model size from approximately 20M to 2B parameters by adjusting the depth and width of the Transformer layers. First, we observe an obvious performance stratification regarding data scalability: models trained on longer historical horizons (represented by cooler colors, *e.g.,* 2015-2016) consistently outperform those trained on limited recent data (warmer colors), validating that extensive historical context helps capture diverse market regimes. Second, regarding model scalability, we observe a growth in performance that saturates at a specific capacity threshold at approximately 10^2M parameters, indicating that the size is sufficiently robust for uni-modal time-series pretraining, balancing high predictive accuracy with computational efficiency before the integration of multimodal reasoning layers.

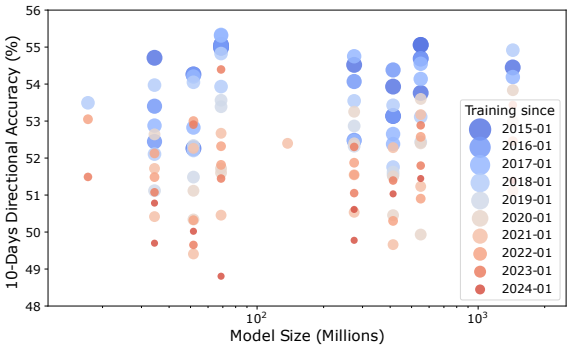


Figure 7: Scalability analysis of pretraining performance w.r.t model size (log scale) and training data.

Table 6: QA performance of **STOCK-R1**-4B with different sizes of time-series encoder.

Setting	Stage	Pure Forecast	Event Detection	Macro Fundamental	News Sentiment	Multi-Signal Reasoning	Overall
4B - Small	SFT	37.04	63.26	50.78	48.99	58.05	52.00
	SFT + u-GRPO	48.11	59.18	48.80	60.09	71.07	57.50
4B - Base	SFT	45.62	65.75	48.61	56.64	65.80	56.50
	SFT + u-GRPO	51.00	68.52	59.70	51.83	63.02	59.00
4B - Large	SFT	41.62	58.12	60.00	57.37	60.36	55.60
	SFT + u-GRPO	50.28	60.22	61.38	55.49	63.72	58.21

Table 6 examines how encoder capacity interacts with the underlying LLM size in downstream financial QA tasks. We do not observe a clear monotonic trend with respect to encoder scale. While moderate encoder capacity performs competitively, further scaling does not consistently translate into additional gains. Overall, these results highlight the importance of considering encoder–LLM capacity matching rather than indiscriminately increasing model size.

C.5 INVESTMENT SIMULATION

Backtest Setting. To assess whether improved forecasting performance translates into economically meaningful decisions, we further evaluate all time-series models in a unified investment simulation. We design a standardized backtesting strategy that converts each model's 10-day close-price forecast into discrete buy or sell signals. At each trading step, the predicted cumulative return is normalized by recent realized volatility to form a z-score signal. When the signal exceeds a fixed threshold, the strategy enters a long or short position with a fixed capital allocation and holds it for a fixed horizon. This protocol enforces identical trading rules across all models, ensuring that return differences stem from forecasting quality rather than strategy engineering.

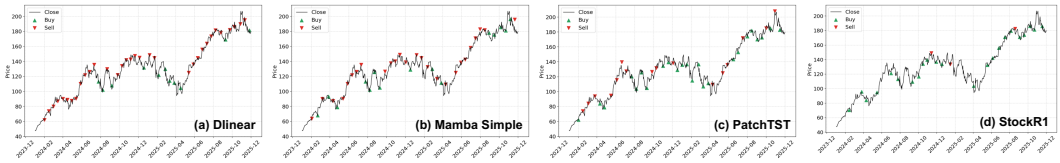
Results. Table 7 reports portfolio-level metrics over U.S. equities spanning multiple market sectors, including technology (*e.g.,* NVDA), finance (*e.g.,* JPM), consumer (*e.g.,* WMT), and broad-market indices (*e.g.,* SPY). Overall, **STOCK-R1** achieves the strongest risk-adjusted performance, with the highest Sharpe ratio and annualized return among all baselines. While exhibiting a higher market beta, **STOCK-R1** maintains a competitive drawdown, indicating that its increased exposure to the market is driven by more confident and selective trend exploitation rather than excessive risk-taking. In contrast, several classical time-series baselines yield unstable or even negative Sharpe ratios.

We also observe distinct behavioral differences in decision patterns across models. For instance, DLinear (Nie, 2022) frequently exhibits a contrarian pattern, tending to go short during rising markets and go long during drawdowns, which leads to systematic trend misalignment. In comparison, **STOCK-R1** produces more market-aware decisions, selectively increasing exposure when predictive signals are strong and remaining conservative under ambiguous conditions. This behavior is reflected

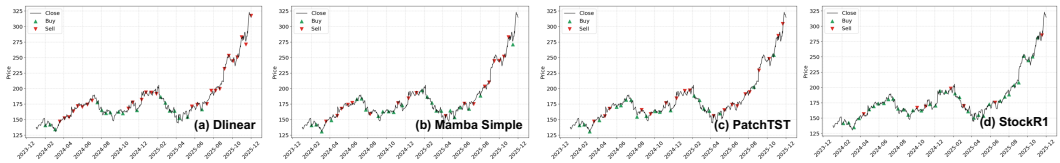
Table 7: Backtesting (from 2025-01-01 to 2025-10-01) performance of forecasting models (**SP**: Sharpe Ratio, **AR**: Annualized Return, **MDD**: Maximum Drawdown, **TT**: Total Traders, **WR**: Win Rate)

Model	SP	AR	MDD	Beta	TT	WR (%)
ARIMA	0.455	-2.78	16.15	0.080	12.33	51.03
Prophet	-0.813	0.56	11.83	0.239	15.56	41.39
NSTransformer	0.414	8.67	8.26	0.204	13.22	48.38
TimesNet	0.027	11.22	7.60	0.280	13.56	47.44
iTransformer	0.873	10.00	8.21	0.308	13.67	**61.21**
PatchTST	0.976	15.67	**7.19**	0.221	13.00	51.25
TimeMixer	0.705	17.78	7.85	0.305	13.11	50.67
DLinear	-0.266	20.78	10.20	0.134	15.11	37.67
Mamba	-0.008	25.22	8.57	0.400	13.33	47.42
STOCK-R1	**1.059**	**31.89**	12.11	**0.521**	14.00	59.48

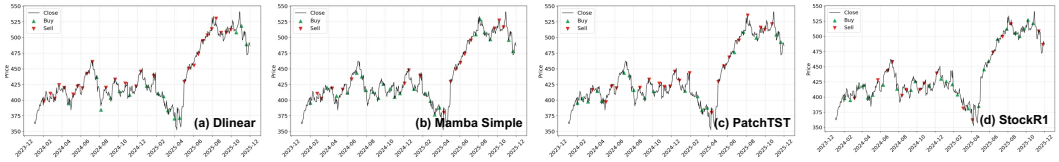
in its higher win rate and superior Sharpe ratio, suggesting that the model predicts more decision-relevant signals beyond short-term price fluctuations.



(1) Directional Trading Signal Prediction for $NVDA



(b) Directional Trading Signal Prediction for $GOOGL



(c) Directional Trading Signal Prediction for $MSFT

Figure 8: Directional Trading Signal Generated by Different Models

Figure 8 provides a qualitative comparison of trading signals generated by different forecasting models under the unified backtesting protocol described in Sec. C.5. Although all models are evaluated using the same signal extraction and execution rules, their resulting buy/sell behaviors exhibit markedly different temporal patterns, reflecting distinct implicit decision logics. Classical linear and decomposition-based models (*e.g.,* DLinear) exhibit highly reactive and oscillatory signals, often trading against prevailing trends, indicating short-horizon extrapolation and poor trend awareness. More expressive neural forecasters (*e.g.,* PatchTST) generate smoother, trend-following signals but tend to maintain persistent exposure across regime shifts due to point-forecast objectives that lack explicit uncertainty modeling. In contrast, **STOCK-R1** produces more selective and market-aware signals, increasing exposure primarily when predictive signals are strong and remaining conservative under ambiguous or volatile conditions. As a result, **STOCK-R1** achieves superior risk-adjusted performance (Table 7), indicating that its advantage stems from improved decision relevance rather than trading strategy engineering. We acknowledge that accurately predicting short-term price movements remains inherently challenging in financial markets; nevertheless, our goal is not perfect

directional accuracy, but to provide more informative and risk-aware buy/sell signals that better support practical decision-making.

C.6 CASE STUDIES

As shown in the case study, the model exhibits strong high-level consistency across its reasoning trace, structured forecast action, and decoded time series. The reasoning identifies sustained selling pressure and repeated failure near the 87 - 88 resistance range, which is reflected in the forecast action as `direction = down`. The time-series decoder further supports this outlook by predicting a decline in closing price by Day 7. Although minor numerical discrepancies may still exist between the qualitative reasoning and decoded trajectory, their overall directional and semantic alignment provides sufficient grounding for the model to reach the correct answer.

A Case Study of Model Internal Consistency

Ticker: SBUX, **Date**: 2025-05-15
Question: Will the predicted 7-day cumulative return stay positive?

Response:

To answer this, I need to first look at and predict how things might evolve in the future.
SBUX has been under sustained selling pressure: from the mid-$90s in late March/early April the stock has declined steadily to the current level around $86, and momentum indicators remain weak. The recent bounce from $81 appears to be a relief rally within a broader downtrend rather than a true reversal, as the stock has failed to reclaim the $87–88 resistance zone on declining volume.

Fundamentally, the turnaround remains uncertain. Q2 2025 results show solid revenue growth but thin margins (7% operating, 4% net) and a highly leveraged balance sheet (liabilities-to-equity ∼-5, current ratio 0.64). The May 12 article highlights "consumer pressure" and "turnaround challenges," and the May 8 piece frames Starbucks as a buy only if investors are willing to wait for a long-term recovery, implying near-term downside risk.

Given this backdrop, the stock is likely to break below current support in the first few sessions, testing the low-$80s to high-$70s area as sellers reassert control. A trough near $77–78 is plausible around mid-window, followed by a partial recovery toward the low-$80s as bargain hunters step in. However, the recovery is unlikely to reclaim the $86 starting level within the forecast window, leaving the cumulative return negative. ▷ *reasoning traces generated by the LLM*

<forecast_action> ▷ *structured forecast generated by the LLM*

```
{future_window:  t+1_t+10
start_value:     86.0    end_value:      83.6
max_value:       84.8    min_value:      77.6
direction:       down      end_change_pct:  -2.79
volatility:      medium    peak_timing:    t+8_t+8
trough_timing:   t+5_t+6    monotonicity:  non_monotonic
mean_close:      81.3      tail_risk:      medium
...  }
```

The time-series decoder forecasts the future 10 days' prices and volume:

<forecast_ts> ▷ *generated by a time-series decoder in a tool-call way*

```
Close price:  [82.36, 79.73, 82.24, 80.74, 78.20, 77.57, 80.97,
84.80, 82.40, 83.63]
High price: [89.23, 84.13, 85.98, ...]
Low price:  [81.68, 72.52, 77.86, ...]
Volume:    [12916896, ...]
```

To determine if the predicted 7-day cumulative return will stay positive, I need to calculate the cumulative return over the next 7 days using the forecasted close prices. The formula for cumulative return is:
((P_{t+7} - P_t) / P_t) * 100, where P_t is today's close and P_{t+7} is the forecasted close 7 days from now.

Today's close: 85.94
Forecasted close prices for the next 7 days: [82.37, 79.73, 82.25, 80.74, 78.20, 77.57, 80.98]

Step 1: Calculate the cumulative return over 7 days:
Day 1: (82.37 - 85.94) / 85.94 = -0.04 (4.16% drop)
Day 2: (79.73 - 82.37) / 82.37 = -0.03 (3.18% drop)
Day 3: (82.25 - 79.73) / 79.73 = 0.03 (3.15% gain)
Day 4: (80.74 - 82.25) / 82.25 = -0.02 (1.84% drop)
Day 5: (78.20 - 80.74) / 80.74 = -0.03 (3.15% drop)
Day 6: (77.57 - 78.20) / 78.20 = -0.01 (0.79% drop)
Day 7: (80.98 - 77.57) / 77.57 = 0.04 (4.41% gain)

Step 2: Calculate the cumulative return after 7 days:

((80.97 - 85.94) / 85.94) * 100 = **-5.78%**

Step 3: Check if the cumulative return is positive:
-5.78% is negative.

Therefore, the predicted 7-day cumulative return will not stay positive.

<answer>**No**</answer>

C.7 Numerical Grounding Evaluation Rubric

We evaluate the numerical grounding quality of model outputs using an LLM-based judge. The judge receives each sample's question, generated reasoning, forecast hint (a structured JSON summary including start/end/max/min values, direction, and percentage change), the full forecast time series, and the model's predicted answer. Crucially, the judge is instructed to assess internal numerical faithfulness. The rubric defines four metrics, each scored continuously in $[0, 1]$:

- **Answer–Evidence Consistency (M1).** Whether the final answer is consistent with the numerical evidence in the forecast time series and hint.
- **Trajectory–Action Alignment (M2).** A field-level match ratio between the forecast action and the generated time series. The judge checks whether summary statistics (*e.g.,* start value, end value, max, min, direction, percentage change) faithfully reflect the underlying price arrays, with a small rounding tolerance (≤ 0.02).
- **Logical Validity (M3).** Whether the final answer, forecast action, generated time series, and reasoning are mutually consistent.
- **Action Success Rate (M4).** Whether the model successfully generates a complete, executable forecast action with all required fields, assessing structural completeness rather than correctness.

The full evaluation prompt, including detailed scoring guidelines and output format specifications, is available in our codebase.

D Additional Experimental Controls

The following tables report additional recorded controls at the 4B scale. Accuracy entries are percentages; trajectory–action entries are reported alignment scores on a 0–100 scale. These results lack per-run predictions, seed information, and confidence intervals in the available records, so small differences should not be treated as statistically established. They are existing results, not experiments newly run for this revision.

D.1 IDENTITY-MASKED INFERENCE

Ticker names, company names, and absolute dates are replaced with anonymous identifiers at inference, preserving the underlying evidence and temporal order. The model is not retrained in this control. The observed changes are -0.20, $+0.24$, and -0.3 percentage points, respectively. Without uncertainty estimates, these point differences do not establish statistical equivalence. Market paths and news can retain identifying information even after explicit names and dates are removed.

Table 8: Identity masking. Near-unchanged scores constrain one inference-time lookup mechanism, but do not exclude teacher-side hindsight or implicit event identification.

Setting	Forecast QA	Avg. QA	10-day accuracy
STOCK-R1-4B	51.0	60.8	59.6
Identity-masked	50.8	61.0	59.3

D.2 PROMPTING AND UNCONDITIONED-TOOL CONTROLS

Direct prompting, chain-of-thought (CoT), few-shot prompting, and access to an unconditioned time-series tool are compared in Table 9. "Direct" denotes the recorded baseline without the added CoT or few-shot prompt intervention; it does not imply that a model such as o3 has no native reasoning mechanism. These variants test access to structured numerical evidence as well as prompting. Exact prompts, demonstration selection, and decoding budgets were not retained in the supplied records; the comparisons do not establish equal inference compute or optimized prompting for every model. Prompting improves Forecast QA, while a numerical tool gives larger gains in these runs. This supports forecasting-tool access as a substantial contributor to the original gap. It does not show that prompting is unimportant or that the remaining gap is entirely caused by action conditioning.

Table 9: Stronger prompting and tool access. The Qwen3 tool baseline here is a prompting/interface control, distinct from the separate tool-interface comparison in Table 10.

Model	Setting	Forecast QA	Avg. QA
Qwen3-4B	Direct	7.9	43.0
	+ CoT	11.9	45.4
	+ Few-shot	14.8	47.2
	+ Unconditioned TS tool	41.5	55.6
GPT-5	Direct	8.1	46.7
	+ CoT	12.6	48.9
	+ Few-shot	16.1	50.8
	+ Unconditioned TS tool	43.0	56.8
OpenAI o3	Direct	2.8	47.8
	+ CoT	5.6	49.2
	+ Few-shot	9.8	51.3
	+ Unconditioned TS tool	44.1	57.4
STOCK-R1-4B	**Action-conditioned tool**	**51.0**	**60.8**

D.3 UNCONDITIONED VERSUS ACTION-CONDITIONED TOOL

The recorded interface comparison holds the language model, forecasting-module family, inputs, and evaluation protocol fixed, and compares $p_\phi(y \mid \mathbf{H}_{\text{ts}})$ with $p_\phi(y \mid \mathbf{H}_{\text{ts}}, e_\psi(a))$. The records do not establish whether the unconditioned arm was retrained with a matched action-free SFT/RL protocol or formed by removing conditioning at inference. Action conditioning is associated with a 4.1-point improvement in directional accuracy and a 3.81-point improvement in average QA. This is more directly relevant to the interface than removing time-series tokens, but fully matched retraining and

Table 10: Tool-interface comparison. The unresolved training protocol limits causal attribution to action conditioning alone.

Interface	Return RMSE ↓	10-day acc.	Avg. QA	Traj.–action
Unconditioned TS tool	0.043	55.5	57.0	12.7
Action-conditioned tool	0.040	59.6	60.81	53.8

run-level statistics are needed for a stronger mechanism claim. The 57.0 average here and 55.6 in Table 9 belong to distinct reported controls and are not interchangeable.

D.4 Inference-Time Action-Field Masking

Groups of action fields are masked at inference, without retraining. Numerical summaries include trajectory values and aggregate changes; market-state fields describe direction and risk/range categories; temporal-shape fields describe turning points, timing, and monotonicity. All three groups

Table 11: Action-field masking. A drop reflects sensitivity to missing fields, including possible inference distribution shift; it is not a retraining-based redundancy test.

Action input	10-day acc.	Avg. QA	Traj.–action
Full action	59.6	60.81	53.8
Without numerical summaries	58.9	59.8	46.7
Without market-state fields	56.8	57.7	39.5
Without temporal-shape fields	58.6	59.4	44.8

affect the reported metrics, with the largest drop from masking market states. The exact masking representation and scoring treatment for absent fields are not documented. The study therefore supports field dependence but not a uniquely optimal schema, nonredundancy of every field, or robustness to different thresholds.

E Limitations

This work focuses on daily-resolution equity data from S&P 500 stocks, where aligned OHLCV signals, fundamentals, news, and macroeconomic context are relatively accessible. Extending the framework to higher-frequency trading data, broader asset classes, and longer-horizon macro-financial reasoning is a promising direction for future work. Our current forecast action design captures key trajectory-level attributes, but richer action spaces could further model uncertainty, regime shifts, and multi-asset dependencies. Future work may also explore tighter integration with portfolio-level objectives and adaptive action schemas for different financial tasks.