FinHopper: A Progressive Multi-Hop Benchmark for Complex Financial Agents with Step-wise Evaluation

Yuxuan Wu[1,2,*], Miaopei Lin[2,*], Qiufang Ying[2,*], Lin Dan[2], Hao Zhao[2], Qirui Liu[1,2], Yuteng Shen[2], Shaohui Wu[2], Weihong Luo[2,†], Xiku Du[2], Xing Tang[3,†], Hao Chen[2,†]

[1]South China University of Technology

[2]FiT, Tencent

[3]Shenzhen Technology University

Abstract

Financial agents are expected to answer factual investment queries through open search, but financial multi-hop retrieval differs from general multi-hop QA in a crucial way: each intermediate answer is an exact typed constraint, such as an entity, date, event, or numerical indicator, that determines the next query. This strict answer-conditioned dependency makes small intermediate errors cascade into plausible but wrong search trajectories. We introduce **FinHopper**, a benchmark for complex financial agents with controlled reasoning depth from 1 to 6 hops, covering stocks, funds, and macroeconomic facts, with gold intermediate answers for process-level evaluation. To construct FinHopper at scale, we design a state-transition-based synthesis framework that composes verified atomic financial QA patterns into dependency-preserving chains, followed by targeted in-synthesis verification, model-ensemble checking, and public-search validation by financial experts. Experiments with 20 LLM-agent configurations show that public search tools are necessary but insufficient: performance drops sharply as hop depth increases, the best model remains far below human experts, and process-aware scoring reveals partial reasoning progress hidden by final-answer-only evaluation. The benchmark is available at https://github.com/justajustin/FinHopper/.

1 Introduction

Financial analysts rarely answer investment questions by retrieving a single fact. A typical workflow may first identify a target company from a comparison, use that company to query an event date, and then use the date to retrieve a market indicator or fund movement. LLM-based agents with search and tool-use abilities (Nakano et al., 2021; Yao et al., 2023; Schick et al., 2023) are increasingly expected to automate such evidence collection (Wu et al., 2023; Nie et al., 2024; Tongyi DeepResearch Team et al., 2025). This creates a financial multi-hop retrieval problem in which every hop must recover an exact, typed answer before the next search can even be posed correctly.

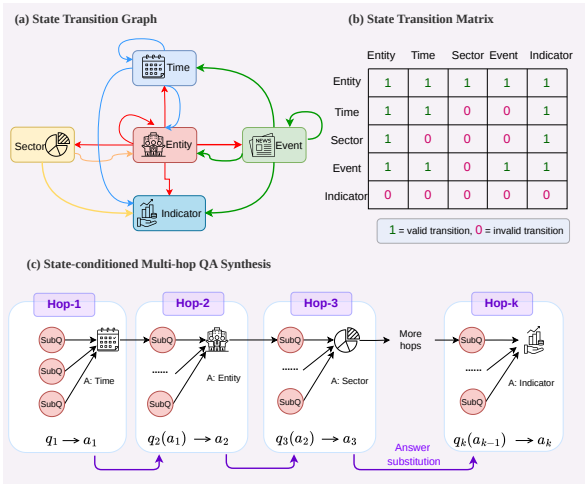


Figure 1: FinHopper abstracts financial multi-hop search as state-conditioned answer transitions, where each verified answer constrains the next hop through dependency-preserving answer substitution.

This structure is different from many general multi-hop QA or browsing tasks. In general domains, hops often gather complementary evidence, and approximate entity matching or broad web exploration can still lead to a usable answer. In finance, however, an intermediate answer becomes a hard query condition: a wrong company, reporting period, event date, or metric definition sends all later searches to the wrong state while the trajectory may still look coherent. Figure 1 abstracts this property as transitions among answer states, where only valid financial answer types can constrain subsequent hops. For example, identifying *"the liquor company with the highest Q3 2024 revenue"* requires exact reported figures and period definitions,

* Equal contribution.

† Corresponding Author.

not a likely or famous brand.

This financial specificity motivates different benchmark design choices from general multi-hop data. A benchmark must control which answer states can be chained, preserve hidden intermediate answers, and verify that every hop remains factually grounded in public financial evidence. Existing financial benchmarks still leave this gap: FinQA (Chen et al., 2021), TAT-QA (Zhu et al., 2021), and ConvFinQA (Chen et al., 2022) emphasize numerical reasoning with provided evidence; FinAgentBench (Choi et al., 2025) and related settings evaluate retrieval or tool use under more constrained formats; and FinSearchComp (Hu et al., 2025) moves closer to open financial search but is still primarily judged by final-answer correctness. Current benchmarks therefore rarely combine public-search reproducibility, strict answer-conditioned chaining, and process-level supervision.

We propose **FinHopper**, a benchmark designed around these financial multi-hop requirements. FinHopper contains questions from 1 to 6 reasoning hops across stocks, funds, and macroeconomic facts. Each item includes a final answer and step-wise gold intermediate answers, allowing evaluation to distinguish agents that fail immediately from those that retrieve correct intermediate evidence but break later in the chain. Its construction mirrors the motivation above: state-conditioned synthesis enforces valid financial dependencies, chain-merge verification prevents answer leakage or ambiguity, and expert public-search validation checks that released step answers are externally reproducible.

Our experiments show that FinHopper acts as a stress test for agentic financial retrieval. Without public search tools, all evaluated models nearly fail. With tools, performance improves substantially but still drops sharply as hop depth increases, revealing model-specific capability boundaries. Process-aware scoring further shows that some agents can make partial progress even when their final answers are wrong, which final-answer-only metrics would hide.

Our contributions are as follows:

1. **A benchmark for dependency-preserving financial search.** We introduce FinHopper, a multi-hop financial benchmark with controlled depth from 1 to 6 hops and hybrid coverage across stocks, funds, and macroeconomics.
2. **A scalable and verified construction framework.** We propose a state-transition-based synthesis method that composes atomic financial QA patterns into valid reasoning chains, with verification at both single-hop and chain-merge stages.
3. **A diagnostic evaluation protocol.** We provide step-wise gold answers and a process-aware scoring metric that localizes failures and exposes partial progress beyond binary final-answer accuracy.

2 Related Work

2.1 Multi-Hop QA and Web Agent Benchmarks

Multi-hop QA and browsing benchmarks have driven progress in evidence retrieval, long-context reasoning, and tool-augmented interaction. HotpotQA (Yang et al., 2018), FEVER (Thorne et al., 2018), Wizard of Wikipedia (Dinan et al., 2019), ELI5 (Fan et al., 2019), and KILT (Petroni et al., 2021) evaluate knowledge-intensive QA over retrieved or provided evidence. WebShop (Yao et al., 2022), WebArena (Zhou et al., 2024), Mind2Web (Deng et al., 2023), GAIA (Mialon et al., 2024), and BrowseComp variants (Wei et al., 2025; Zhou et al., 2025; Chen et al., 2026) further test web interaction and long-chain search. These benchmarks are not centered on financial answer-conditioned dependencies: many tasks reward successful navigation or approximate evidence gathering, whereas financial multi-hop retrieval requires exact entities, dates, metric definitions, and numerical values whose errors cascade across later hops.

2.2 Financial QA and Search Benchmarks

Financial benchmarks cover numerical reasoning, document understanding, and agentic decision making, but they do not fully capture open-domain sequential retrieval. FinQA (Chen et al., 2021), TAT-QA (Zhu et al., 2021), and ConvFinQA (Chen et al., 2022) focus on numerical reasoning over provided documents. FinAgent (Zhang et al., 2024) and MarS (Li et al., 2025) study LLM-based stock-market decision making with financial signals, while FinAgentBench (Choi et al., 2025) introduces tool-augmented financial tasks under a more constrained retrieval setting. FinSearchComp (Hu et al., 2025) moves closer to open-domain financial search, but its complex setting is still primarily assessed with final-answer pass/fail outcomes. Some

benchmarks provide rationales or step annotations, and Finance Agent Benchmark (Bigeard et al., 2025) includes step-wise information, but they do not make verified intermediate answers central to benchmark construction and scoring. FinHopper instead targets public financial search where each intermediate answer becomes a typed condition for the next query, and its process-aware scoring diagnoses whether an agent fails in retrieval, state tracking, chain merging, or final synthesis.

3 Methodology

Building on Figure 1, FinHopper composes verified atomic financial questions into dependency-preserving chains. Figure 2 summarizes the pipeline: state-conditioned pattern sampling, single-hop QA generation and validation, and chain merge with post-synthesis validation before release.

3.1 Task Formulation

A FinHopper instance is a K-hop financial QA chain $\{(q_1,a_1),\ldots,(q_K,a_K)\}$, where q_K is the released user-facing question and a_K is the final answer. The intermediate answers $\{a_1,\ldots,a_{K-1}\}$ are not exposed to the model during evaluation, but they are retained as gold process annotations. The defining property is strict dependency: answer a_t must provide a concrete constraint, such as an entity, date, sector, event, or indicator, that is required to formulate or solve the next hop q_{t+1}.

We distinguish three levels of granularity. A **simple question** queries one elementary financial relation, such as an entity-time-indicator fact or an event date. A **single-hop question** may contain multiple simple sub-queries that can be solved in parallel, such as comparing several entities under the same time and indicator. A **multi-hop question** links single-hop questions sequentially through answer substitution, so that each new hop depends on the verified answer of the previous hop.

3.2 Atomic Financial Patterns

To make synthesis controllable, we define a pattern library for generating single-hop QA pairs. Each pattern specifies the input components, the required computation, and the answer type. Across financial domains, these components can be represented by five answer states:

$$\mathrm{S} = \{\text{Entity, Time, Sector, Event, Indicator}\}.$$

We design 21 patterns covering entity-time comparison, event-based retrieval, and entity-relationship queries. Pattern variants combine the question component type, computation type (e.g., ranking, filtering, comparison, or extraction), and answer component type, which determines how the next hop can be sampled. The full taxonomy is provided in Appendix E.

3.3 State-Conditioned Chain Synthesis

Not every answer type can validly constrain every subsequent question. For example, an entity answer can lead to questions about its sector, event timeline, or indicators, while an indicator value usually functions as a terminal answer. We therefore model chain construction as a random walk over an answer-state transition graph.

We define a state transition graph $G = (\mathrm{S},\mathrm{E})$, where each directed edge $(s_i \to s_j) \in \mathrm{E}$ indicates that an answer of state s_i can condition a following question whose answer state is s_j, as shown in Figure 1. The transition probability from state s_i to s_j is defined by the normalized transition matrix $\mathbf{P} \in \{0,1\}^{|\mathrm{S}|\times|\mathrm{S}|}$:

$$P_{ij} = \begin{cases} \frac{1}{|D(s_i)|} & \text{if } (s_i \to s_j) \in \mathrm{E} \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

where $D(s_i) = \{s_j : (s_i \to s_j) \in \mathrm{E}\}$ is the set of valid outgoing states. Starting from an initial state $s_0 \sim \pi_0$, we sample a target depth K and generate a valid trajectory $(s_0,\ldots,s_K)$ using $\mathbf{P}$. Each sampled state selects an eligible pattern and data pool; after verification, its answer is substituted into the next hop as an explicit constraint. Detailed transition examples are listed in Table 6.

3.4 Chain Generation and Merge

3.4.1 Single-Hop Generation

Given a sampled pattern, an LLM generator equipped with internal financial database tools retrieves candidate factual evidence and instantiates a single-hop QA pair. Each pair must pass single-hop verification before it can be merged into a longer chain.

3.4.2 Chain Merge

After a single-hop QA passes verification, it is merged with the existing chain through **answer substitution**: the previous hop's answer is embedded into the next hop as a required constraint. The released question hides intermediate answers,

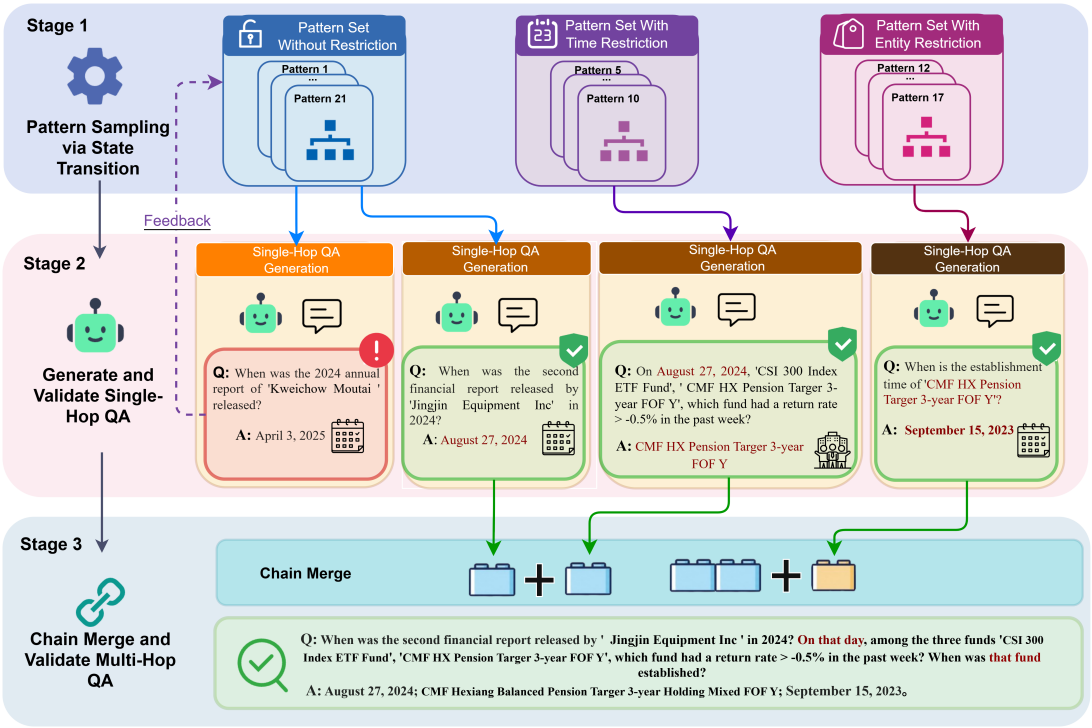


Figure 2: Fine-grained construction pipeline of FinHopper. The framework first samples reusable financial patterns according to valid answer-state transitions, then generates and validates single-hop QA pairs, and finally merges verified hops into multi-hop questions while preserving intermediate answers for process-level evaluation.

while the benchmark stores them as process-level gold references. Chain-merge verification ensures that the new question is self-contained, unambiguous, and faithful to the underlying evidence.

3.5 Quality Control

FinHopper uses quality control at two stages: in-synthesis verification filters errors during generation, and post-synthesis verification validates released items independently from the construction process.

3.5.1 In-Synthesis Verification

In-synthesis verification checks both atomic QA pairs and chain-merge operations before they enter the final benchmark, preventing factual errors from accumulating across hops. The detailed verification flow is shown in Appendix A.

Single-Hop Verification Single-hop verification re-derives tool parameters from the question text alone, retrieves evidence, and checks answer consistency. Failed or ambiguous samples are regenerated rather than merged.

Chain-Merge Verification For chain merge, separate critics check the question and answer: the question critic prevents leakage, ambiguity, and logical drift, while the answer critic checks completeness and faithfulness to evidence. We sample parallel merge candidates and retain the best validated one. Detailed criteria and prompts are provided in Appendices F and J.

3.5.2 Post-Synthesis Verification

After synthesis, we apply two independent checks before releasing the benchmark: model-ensemble answer verification and human verification through public search.

Answer Correctness Verification We employ an ensemble of five frontier LLMs under a ReAct framework (Yao et al., 2023). At this construction-verification stage, models query the internal financial database to derive reasoning chains and final answers. A sample is accepted only when a strict majority (≥ 3 agents) agree; disagreements are sent to expert arbitration. Implementation details are provided in Appendix H.

Human Verification As a final safeguard, 16 financial experts verified every released step-level answer through public search engines rather than the internal database. We also sampled 30% of synthesized items from each complexity level for double-blind review and senior-expert adjudication.

Table 1: Hop-wise composition of FinHopper. Each K-hop item contains K step-level gold answers, yielding 572 step-level checking points in total. These step-level annotations are diagnostic signals rather than independent test samples.

Hop	1	2	3	4	5	6
Level	L1	L1	L2	L2	L3	L3
#Items	50	50	39	21	19	21
#Step answers	50	100	117	84	95	126

The 95% pass rate confirms alignment between automated and expert public-search checks.

4 Experiments

We conduct an evaluation on FinHopper to assess the capability of LLM agents in addressing financial tasks. Our evaluation focuses on four key aspects: the necessity of tools, the human-AI performance gap, the impact of reasoning depth, and inference efficiency.

4.1 Benchmark Statistics

Figure 3 summarizes FinHopper. The benchmark contains 200 questions with a controlled difficulty gradient: L1 covers 1–2 hops, L2 covers 3–4 hops, and L3 covers 5–6 hops. As difficulty increases, question components also become denser: L3 questions involve 2.7× more entities and 2.9× more time references than L1. The benchmark spans 14 financial domains, including intraday data, cross-period analysis, individual stocks, fund metrics, and macroeconomics. FinHopper includes Chinese and English versions with the same reasoning chains and step-level gold answers; the main large-scale evaluation uses the Chinese version, with representative English results in Section 4.5.

4.2 Evaluation of FinHopper

Financial tasks require multi-hop reasoning, so binary final-answer metrics cannot capture partial progress or hallucinated trajectories. We therefore use a two-stage hierarchical score. First, we set $s_{final} = 1$ if the generated final answer matches the gold answer after normalizing units, dates, and formatting; otherwise $s_{final} = 0$. If the final answer is wrong, we evaluate whether the response recovers the N gold intermediate answers $\mathrm{K} = \{k_1, k_2, ..., k_N\}$:

$$s_{process} = \alpha \cdot \frac{1}{N} \sum_{i=1}^{N} I(k_i \preceq \mathrm{A})$$

where $k_i \preceq \mathrm{A}$ means that the response entails the gold intermediate step k_i, and $\alpha = 0.5$ caps partial credit for incorrect final answers. The final score is:

$$S(\mathrm{A}) = s_{final} + (1 - s_{final}) \cdot s_{process}$$

Correct final answers receive 1.0, while incorrect final answers receive 0.0–0.5 based on process validity. Evaluation prompts are provided in Appendix I.

4.3 Experimental Setup

We evaluate 10 representative frontier model families on the Chinese version of FinHopper: Gemini 3 Pro (Google, 2025), Claude 4.5 Sonnet (Anthropic, 2025), Doubao-seed-1-8-251215 (Bytedance Seed, 2025), GPT-5.2 (OpenAI, 2025), DeepSeek-V3.2 (DeepSeek-AI et al., 2025), HunYuan-2.0 (Tencent Cloud, 2026), Kimi K2 and K2 Thinking (Kimi Team et al., 2025; Kimi Team, 2025), GLM-4.7 (Z.ai, 2025), qwen3-max-2025-09-23 (Alibaba Cloud, 2025), and Grok 4.1 (xAI, 2025). Models answer with public search engines under the ReAct framework (Yao et al., 2023), with a maximum iteration budget of 42; the internal financial database is used only for data construction and is never exposed during evaluation. We test both "thinking" and "no-thinking" modes, and use 20 independent financial experts under the same public-search setting as the human upper bound. ReAct prompts are provided in Appendix H.

Because simple questions (1–2 hops) constitute 50% of the data, we report two metrics to avoid biasing results.

Micro-Average Score: This measures overall performance weighted by the data distribution. Let $Score_k$ denote the score at hop k, and w_k be the proportion of k-hop questions:

$$Score_{\text{micro-avg}} = \sum_{k=1}^{6} w_k \cdot Score_k$$

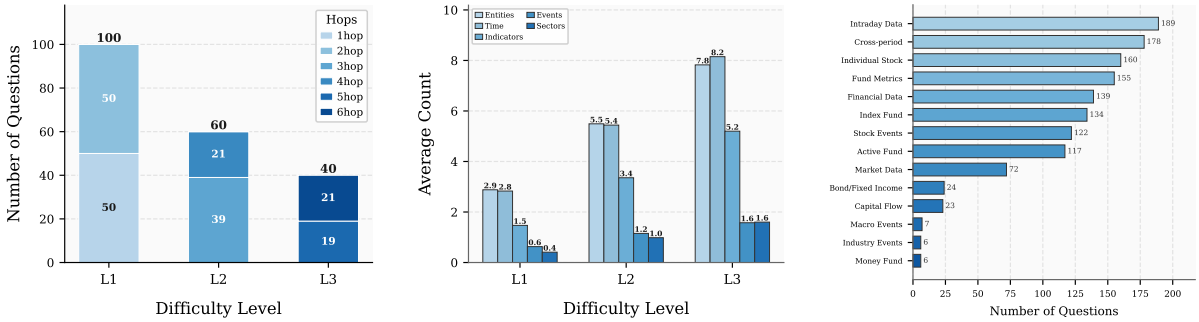
Macro-Average Score: This treats all hop depths equally and better reflects robustness on long-chain tasks:

$$Score_{\text{macro-avg}} = \frac{1}{6} \sum_{k=1}^{6} Score_k$$

We also measure average inference latency in seconds per query.

4.4 Main Result

We report the overall performance on the Chinese version in Table 2, with English-version representative results provided in Table 3. A sharp contrast



(a) Difficulty & Hops (b) Question Complexity (c) Domain Distribution

Figure 3: FinHopper provides a controlled difficulty gradient: hop depth increases from L1 to L3, question components grow denser with difficulty, and items cover diverse financial domains across stocks, funds, and macroeconomics.

Table 2: Performance of various models and human on FinHopper. All scores are reported in %.

model	without tools		use tools								
	Mic Avg.	Time	1hop	2hop	3hop	4hop	5hop	6hop	Mic Avg.	Mac Avg.	Time
Human Performance	0	5	94.0	90.0	84.6	73.4	65.9	64.4	81.7	78.7	324
Claude-4.5-Sonnet-thinking	2.5	18	81.3	**72.5**	**70.2**	63.5	**54.1**	31.1	**67.3**	**62.1**	31
Claude-4.5-Sonnet-no-thinking	1.0	10	79.6	70.7	63.2	60.0	49.6	25.6	63.7	58.1	**24**
Gemini-3-Pro-thinking	3.2	18	80.0	69.1	69.5	60.1	52.2	26.5	65.0	59.6	34
Gemini-3-Pro-no-thinking	2.8	14	81.2	70.3	65.5	53.0	44.2	20.2	62.7	55.7	25
GPT-5.2-thinking	4.2	30	83.4	69.6	56.0	53.4	45.8	19.2	61.3	54.6	121
GPT-5.2	1.4	27	86.8	64.4	50.2	49.2	44.7	12.5	58.5	51.3	108
Doubao-seed-1-8-251215-thinking	1.8	15	79.0	61.3	60.2	**63.9**	46.6	28.8	61.1	56.6	34
Doubao-seed-1-8-251215	3.2	14	78.1	63.4	58.4	57.0	41.2	23.2	59.2	53.6	30
DeepSeek-V3.2-thinking	2.0	24	86.3	72.3	65.7	34.8	10.7	9.3	58.1	46.5	42
DeepSeek-V3.2-no-thinking	4.2	19	**88.5**	69.2	63.1	34.5	9.2	8.0	57.1	45.4	34
Kimi-k2-thinking	3.4	23	67.2	58.5	61.1	53.2	48.2	**38.0**	57.5	54.4	203
Kimi-k2	7.0	20	68.0	50.9	59.7	49.6	44.0	36.1	54.6	51.4	187
HunYuan-2.0-thinking	4.0	26	80.1	60.9	55.5	51.5	31.5	26.4	57.3	51.0	46
HunYuan-2.0-instruct	3.2	19	76.5	58.0	49.2	47.5	24.8	20.0	52.7	46.0	41
Grok-4.1-thinking	3.0	16	66.2	57.7	53.0	50.3	35.3	25.8	52.7	48.1	57
Grok 4.1	6.2	14	63.0	51.9	49.5	46.1	27.0	18.7	42.7	42.7	42
Qwen-max-20250923-thinking	3.2	14	63.0	54.7	53.8	55.9	34.5	32.4	52.5	49.1	47
Qwen-max-20250923-no-thinking	1.0	**6**	61.3	44.5	53.4	50.8	30.6	30.5	48.3	45.2	30
GLM-4.7-thinking	8.1	71	76.1	47.4	50.7	43.6	22.7	14.2	49.0	42.5	92
GLM-4.7-no-thinking	6.4	57	75.2	45.5	45.3	38.5	18.7	11.3	48.8	40.9	73

exists between the "without tools" and "use tools" settings. Without access to public search tools, all models fail almost completely, scoring below 8.1%. With search-tool augmentation, Claude-4.5-Sonnet-thinking achieves the top Micro Average score of 67.3%, but there remains **a significant gap to human experts** (81.7%). This gap expands as the question difficulty increases. For instance, on complex 6-hop tasks, humans maintain 64.4% accuracy, while the best model achieves only 31.1%. A detailed case study is provided in Appendix F.1. To assess the stability of the reported results under repeated evaluations, we report mean and standard deviation for both Micro- and Macro-Average scores in Appendix C.2. The standard deviations remain within 0.7–1.8 points, indicating that the main ranking trends are stable despite the moderate benchmark size.

Furthermore, we present three key findings:

(1) Each model has a specific "agentic capability boundary". We observe a sharp cliff-like performance decline for most models once the question depth reaches a certain hop. As shown in Figure 4, DeepSeek-V3.2 performs well on simple tasks but the score drops by half after 4-hop. Similarly, Gemini-3-Pro maintains high accuracy but eventually hits its boundary at 6-hop.

(2) Models with "thinking" capabilities demonstrate superior robustness compared to standard counterparts. Figure 6 illustrates this performance gain. The benefit of the thinking mode becomes more evident as reasoning difficulty in-

Table 3: Performance of representative models on FinHopper (English). All scores are reported in %.

model	without tools		use tools								
	Mic Avg.	Time	1hop	2hop	3hop	4hop	5hop	6hop	Mic Avg.	Mac Avg.	Time
Gemini-3-Pro-thinking	3.5	**17**	81.5	68.3	**70.8**	**60.8**	**51.5**	26.0	**65.2**	**59.8**	**33**
GPT-5.2-thinking	4.8	29	**84.8**	**69.0**	57.2	54.0	45.2	19.8	61.7	55.0	118
GLM-4.7-thinking	7.5	74	75.0	48.2	51.0	42.8	22.2	14.8	48.8	42.3	95
Kimi-k2-thinking	3.0	25	66.0	59.2	60.5	52.5	48.8	**38.2**	57.3	54.2	208

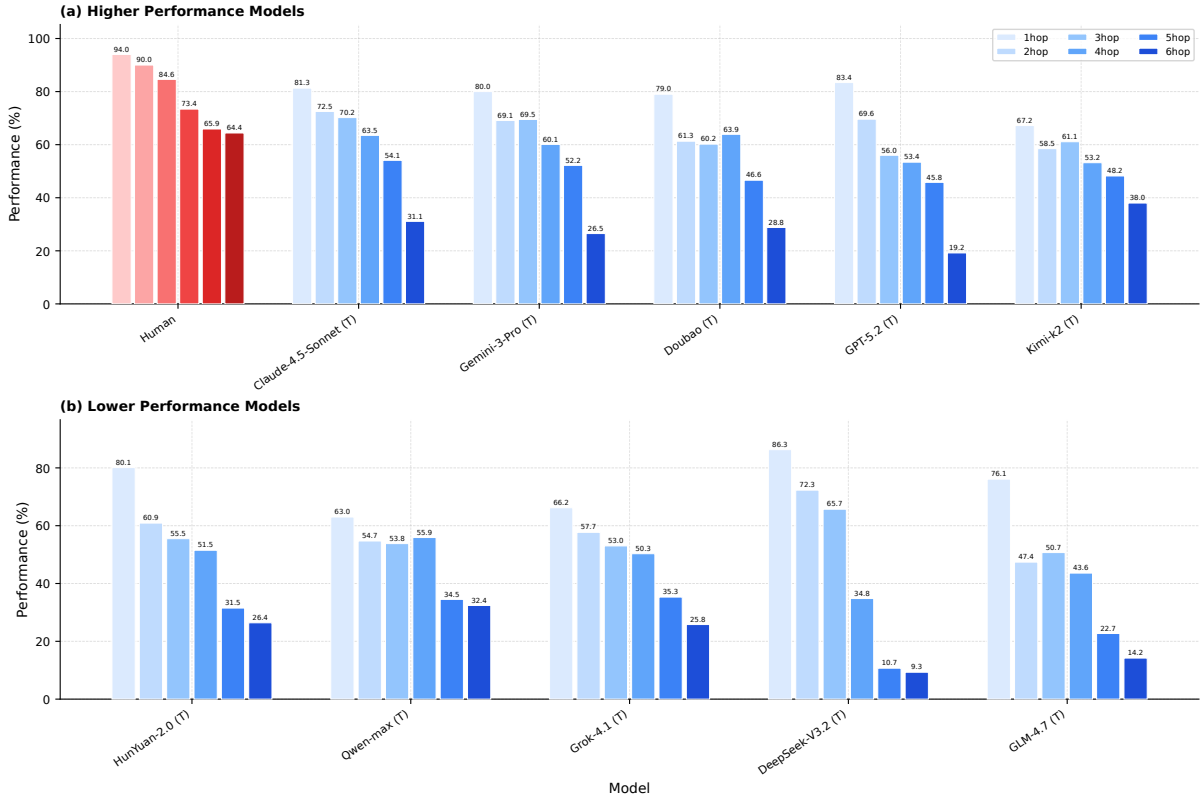


Figure 4: Model performance across 1–6 hop tasks, grouped by thinking mode. The curves expose model-specific capability boundaries: most agents remain competitive on shallow tasks but degrade sharply once the reasoning chain exceeds their effective search and state-tracking depth.

creases. This suggests that the main benefit of reasoning-enhanced inference is not memorizing more financial facts, but maintaining a more reliable execution trace. In long-chain financial search, a model must decompose the user query into ordered sub-goals, preserve typed constraints such as entities, dates, sectors, and indicators, and revise its search plan when an intermediate result conflicts with later evidence. Thinking models appear to be more robust at these planning and state-tracking operations, which explains why the average gain increases from shallow to deeper hops. However, the gain is not uniform across all depths or models: on simple 1–2 hop questions, the additional reasoning budget sometimes provides limited benefit because the main bottleneck is factual retrieval rather than long-horizon control. While the "thinking" mode enhances accuracy, it incurs a computational cost. As shown in the Time column in Table 2, reasoning-enhanced models generally require longer inference times (e.g., GPT-5.2-thinking at 121s vs. GPT-5.2 at 108s).

(3) Step-by-step grading offers more precise assessment of agent capabilities than the binary final answer score , especially for complex tasks (5-6 hops), as shown in Figure 5. A binary score is not able to distinguish answers with different partial success. For instance, on 6-hop tasks, DeepSeek-V3.2 scores only 1.0 on s_{final} but achieves 8.3 on $s_{process}$, indicating the model can solve intermediate steps even if it fails the final check. Thus, step-by-step grading is vital for cap-

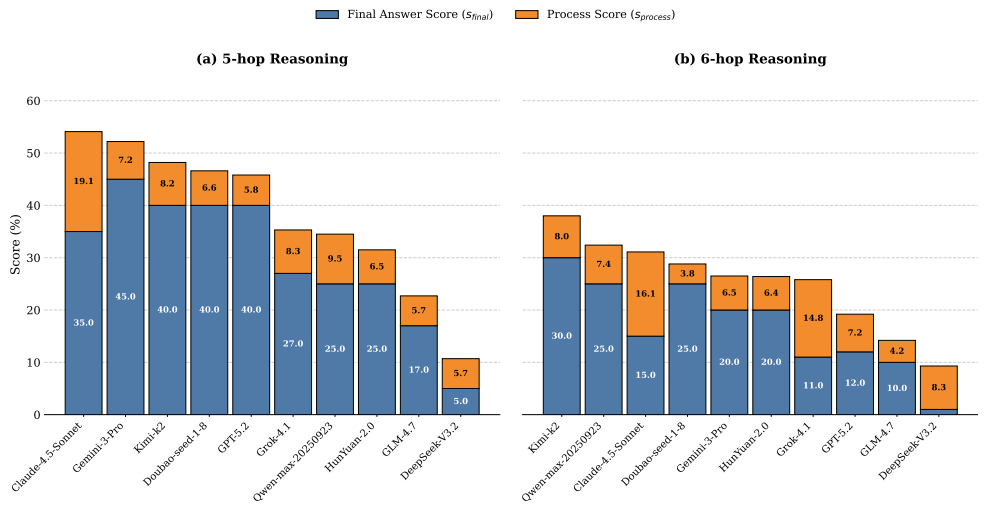


Figure 5: Breakdown of model performance into Final Answer Score(s_{final}) and Process Score ($s_{process}$). The left and right panels show results for 5-hop and 6-hop questions, respectively.

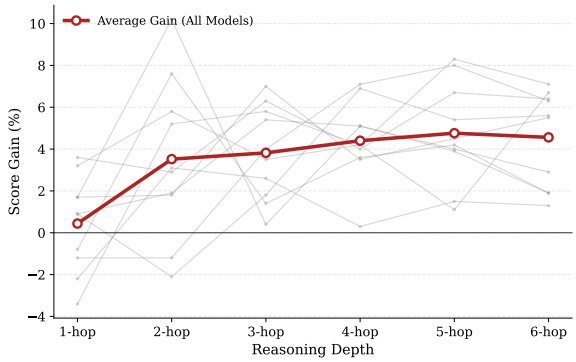


Figure 6: Score gain of Thinking Mode across tasks with depth from 1 to 6 hops. The grey lines represent the score trajectories of individual models, while the bold red line denotes the average gain of all models.

turing the true potential of the agents.

4.5 Representative English-Version Evaluation

FinHopper also includes an English version translated from the Chinese benchmark with the same reasoning chains, final answers, and step-level gold references. Because full tool-augmented evaluation incurs substantial API and search costs, we evaluate the English version on four representative thinking models. As shown in Table 3, the English-version results follow the same broad trend as the Chinese main evaluation, with strong performance on lower-hop tasks and a clear decline as reasoning depth increases.

4.6 Failure Analysis

To better understand the capability boundary revealed by FinHopper, we sample successful and failed model trajectories across difficulty levels and annotate the first major error that causes the reasoning chain to deviate. The failures mainly fall into four categories: decomposition and planning failures (38%), tool misuse (31%), hallucination propagation (17%), and information extraction failures (14%). These patterns suggest that performance drops beyond 4 hops not because models lack isolated financial facts, but because longer chains amplify small errors in planning, tool grounding, evidence extraction, and state tracking into final-answer failures. Detailed examples and qualitative analysis are provided in Appendix F.1.

5 Conclusion

In this paper, we address the lack of benchmarks for agents' capacity in solving complex financial tasks requiring strict sequential dependency and disambiguation of professional metrics by introducing **FinHopper**. It is constructed through a state-transition-based synthesis framework with verified intermediate answers. Our evaluation reveals that while the top-performing model, Claude-4.5-Sonnet-thinking, demonstrates improved robustness, a substantial performance gap remains compared to human experts, particularly on tasks exceeding 4 hops. Furthermore, we provide a step-by-step diagnosis tool with validated intermediate gold reference answers. In future work, we plan to expand the benchmark boundaries by incorporating

broader heterogeneous data sources and synthesizing reasoning chains of higher complexity, continually challenging the limits of agentic reasoning in professional financial domains.

6 Limitations

Despite FinHopper's advancements, limitations remain. **Language and Market Coverage.** Although FinHopper provides both Chinese and English versions, the main large-scale evaluation is conducted on Chinese questions, and the data predominantly stems from the Chinese market, leaving broader generalization to global markets to be further verified. **Naturalness of Queries.** Although auto-generated queries undergo human verification, they still lack the ambiguity and informality typical of real-world questions. **Scope and Modality.** The benchmark focuses on factual reasoning, currently excluding subjective tasks like sentiment analysis or multi-modal interpretation.

7 Ethical Review

Data Collection and Privacy. FinHopper is constructed solely from publicly available sources (regulatory filings, market data, official disclosures) and contains no personally identifiable information (PII). The released benchmark contains the query, final gold answer, and step-wise intermediate answers required for process evaluation. All collection processes comply with respective terms of service and data protection regulations.

Intended Use and Potential Misuse. Designed for evaluating AI reasoning, FinHopper mitigates misuse risks (e.g., market manipulation) by focusing on historical data and reasoning patterns rather than real-time trading signals. We advocate for responsible research in compliance with financial regulations.

Bias Considerations. To address potential biases in financial data (e.g., over-representation of large-cap firms), we employed stratified sampling across company sizes, sectors, and time periods. Users should consider these inherent limitations when interpreting results.

Human Evaluation. Expert evaluations were conducted with informed consent from voluntary participants. No sensitive personal data was collected during the review process.

References

Alibaba Cloud. 2025. qwen3-max model info. Official documentation for the qwen3-max-2025-09-23 snapshot.

Anthropic. 2025. Introducing claude sonnet 4.5. Anthropic News.

Antoine Bigeard, Langston Nashold, Rayan Krishnan, and Shirley Wu. 2025. Finance agent benchmark: Benchmarking llms on real-world financial research tasks. *arXiv preprint arXiv:2508.00828*.

Bytedance Seed. 2025. Seed1.8 model card: Towards generalized real-world agency. Model card, ByteDance.

Zhiyu Chen, Wenhu Chen, Charese Smiley, Sameena Shah, Iana Borova, Dylan Langdon, Reema Moussa, Matt Beane, Ting-Hao Huang, Bryan Routledge, and William Yang Wang. 2021. FinQA: A dataset of numerical reasoning over financial data. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3697–3711.

Zhiyu Chen, Shiyang Li, Charese Smiley, Zhiqiang Ma, Sameena Shah, and William Yang Wang. 2022. Convfinqa: Exploring the chain of numerical reasoning in conversational finance question answering. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6279–6292.

Zijian Chen, Xueguang Ma, Shengyao Zhuang, Ping Nie, Kai Zou, Sahel Sharifymoghaddam, Andrew Liu, Joshua Green, Kshama Patel, Ruoxi Meng, Mingyi Su, Yanxi Li, Haoran Hong, Xinyu Shi, Xuye Liu, Hosna Oyarhoseini, Nandan Thakur, Crystina Zhang, Luyu Gao, and 2 others. 2026. BrowseComp-Plus: A fair and disentangled evaluation benchmark for deep search agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 22349–22370, San Diego, California, United States. Association for Computational Linguistics.

Chanyeol Choi, Jihoon Kwon, Alejandro Lopez-Lira, Chaewoon Kim, Minjae Kim, Juneha Hwang, Jaeseon Ha, Hojun Choi, Suyeol Yun, Yongjin Kim, and Yongjae Lee. 2025. FinAgentBench: A benchmark dataset for agentic retrieval in financial question answering. In *Proceedings of the 6th ACM International Conference on AI in Finance (ICAIF)*, pages 632–637.

DeepSeek-AI, Aixin Liu, Aoxue Mei, Bangcai Lin, Bing Xue, Bingxuan Wang, Bingzheng Xu, Bochao Wu, Bowei Zhang, Chaofan Lin, and 1 others. 2025. Deepseek-v3.2: Pushing the frontier of open large language models. *arXiv preprint arXiv:2512.02556*.

Xiang Deng, Yu Gu, Boyuan Zheng, Shijie Chen, Samuel Stevens, Boshi Wang, Huan Sun, and Yu Su. 2023. Mind2web: Towards a generalist agent for the

web. In *Advances in Neural Information Processing Systems*, volume 36.

Emily Dinan, Stephen Roller, Kurt Shuster, Angela Fan, Michael Auli, and Jason Weston. 2019. Wizard of wikipedia: Knowledge-powered conversational agents. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Angela Fan, Yacine Jernite, Ethan Perez, David Grangier, Jason Weston, and Michael Auli. 2019. Eli5: Long form question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3558–3567.

Google. 2025. A new era of intelligence with Gemini 3. Google Blog, November 18, 2025.

Liang Hu, Jianpeng Jiao, Jiashuo Liu, Yanle Ren, Zhoufutu Wen, Kaiyuan Zhang, Xuanliang Zhang, Xiang Gao, Tianci He, Fei Hu, Yali Liao, Zaiyuan Wang, Chenghao Yang, Qianyu Yang, Mingren Yin, Zhiyuan Zeng, Ge Zhang, Xinyi Zhang, Xiying Zhao, and 4 others. 2025. FinSearchComp: Towards a realistic, expert-level evaluation of financial search and reasoning. *arXiv preprint arXiv:2509.13160*.

Kimi Team. 2025. Introducing Kimi K2 Thinking. Official release, November 6, 2025.

Kimi Team, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru Chen, Yuankun Chen, Yutian Chen, Zhuofu Chen, and 1 others. 2025. Kimi k2: Open agentic intelligence. *arXiv preprint arXiv:2507.20534*.

Junjie Li, Yang Liu, Weiqing Liu, Shikai Fang, Lewen Wang, Chang Xu, and Jiang Bian. 2025. MarS: A financial market simulation engine powered by generative foundation model. In *The Thirteenth International Conference on Learning Representations (ICLR)*.

Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. 2024. GAIA: A benchmark for general ai assistants. In *The Twelfth International Conference on Learning Representations (ICLR)*.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, and 1 others. 2021. Webgpt: Browser-assisted question-answering with human feedback. *arXiv preprint arXiv:2112.09332*.

Yuqi Nie, Yaxuan Kong, Xiaowen Dong, John M Mulvey, H Vincent Poor, Qingsong Wen, and Stefan Zohren. 2024. A survey of large language models for financial applications: Progress, prospects and challenges. *arXiv preprint arXiv:2406.11903*.

OpenAI. 2025. Introducing GPT-5.2. OpenAI Official Website, December 11, 2025.

Fabio Petroni, Aleksandra Piktus, Angela Fan, Patrick Lewis, Majid Yazdani, Nicola De Cao, James Thorne, Yacine Jernite, Vladimir Karpukhin, Jean Maillard, Vassilis Plachouras, Tim Rocktäschel, and Sebastian Riedel. 2021. KILT: a benchmark for knowledge intensive language tasks. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2523–2544. Association for Computational Linguistics.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36:68539–68551.

Tencent Cloud. 2026. Tencent Hunyuan. Official documentation listing the hunyuan-2.0-thinking-20251109 and hunyuan-2.0-instruct-20251111 model identifiers.

James Thorne, Andreas Vlachos, Christos Christodoulopoulos, and Arpit Mittal. 2018. Fever: a large-scale dataset for fact extraction and verification. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 809–819.

Tongyi DeepResearch Team, Baixuan Li, Bo Zhang, Dingchu Zhang, Fei Huang, Guangyu Li, Guoxin Chen, Huifeng Yin, Jialong Wu, Jingren Zhou, and 1 others. 2025. Tongyi deepresearch technical report. *arXiv preprint arXiv:2510.24701*.

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. 2025. BrowseComp: A simple yet challenging benchmark for browsing agents. *arXiv preprint arXiv:2504.12516*.

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

xAI. 2025. Grok 4.1 model card. Model card, xAI.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D Manning. 2018. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2369–2380.

Shunyu Yao, Howard Chen, John Yang, and Karthik Narasimhan. 2022. Webshop: Towards scalable real-world web interaction with grounded language agents. In *Advances in Neural Information Processing Systems*, volume 35, pages 20744–20757.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023. React: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*.

Z.ai. 2025. GLM-4.7: Advancing the coding capability. Official technical blog, December 22, 2025.

Wentao Zhang, Lingxuan Zhao, Haochong Xia, Shuo Sun, Jiaze Sun, Molei Qin, Xinyi Li, Yuqing Zhao, Yilei Zhao, Xinyu Cai, Longtao Zheng, Xinrun Wang, and Bo An. 2024. A multimodal foundation agent for financial trading: Tool-augmented, diversified, and generalist. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 4314–4325.

Peilin Zhou, Bruce Leon, Xiang Ying, Can Zhang, Yifan Shao, Qichen Ye, Dading Chong, Zhiling Jin, Chenxuan Xie, Meng Cao, Yuxin Gu, Sixin Hong, Jing Ren, Jian Chen, Chao Liu, and Yining Hua. 2025. BrowseComp-ZH: Benchmarking web browsing ability of large language models in chinese. *arXiv preprint arXiv:2504.19314*.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. 2024. Webarena: A realistic web environment for building autonomous agents. In *The Twelfth International Conference on Learning Representations*.

Fengbin Zhu, Wenqiang Lei, Youcheng Huang, Chao Wang, Shuo Zhang, Jiancheng Lv, Fuli Feng, and Tat-Seng Chua. 2021. TAT-QA: A question answering benchmark on a hybrid of tabular and textual content in finance. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 3277–3287.

A In-Synthesis Verification Details

Figure 7 illustrates the two filters used during benchmark construction. Single-hop verification independently re-derives answers from question text, while chain-merge verification critiques candidate merged questions and answers before accepting a longer QA chain.

For single-hop verification, the verifier receives only the generated question, re-extracts tool parameters, invokes the corresponding financial tools, and checks whether the generated answer is semantically consistent with retrieved evidence. Samples whose required parameters cannot be reliably extracted are treated as ambiguous and regenerated. For chain-merge verification, the question critic checks process concealment, intent preservation, disambiguation, closure, and logical consistency, while the answer critic checks evidence restriction, completeness, and faithfulness. These checks target the key risks of financial multi-hop synthesis: leaking intermediate answers, drifting away from the original constraint, or propagating unsupported evidence across hops.

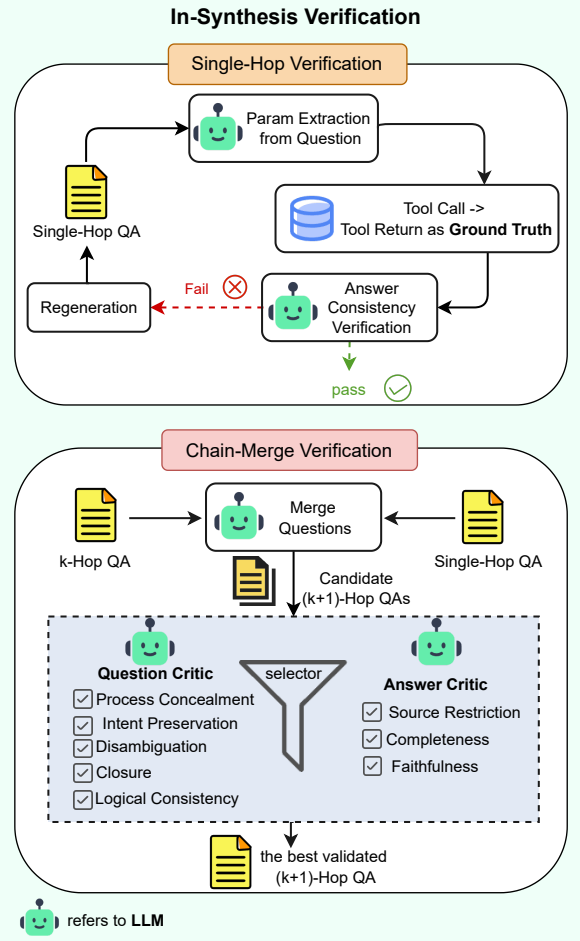


Figure 7: In-synthesis verification independently re-derives single-hop answers and critiques chain-merged questions before they are accepted into longer FinHopper instances.

B Data Pool Definitions

As illustrated in Table 5, the input states are categorized into four distinct dimensions: *Entity*, *Time*, *Sector*, and *Event*.

C Experiment Details on FinHopper

C.1 Model Performance with Increasing Task Difficulty

The detailed 1–6 hop performance trends are discussed in the main text alongside Figure 4.

C.2 Stability Analysis

Table 4 reports the mean and standard deviation of Micro- and Macro-Average scores under repeated evaluations. The overall fluctuations are small, supporting the stability of the main conclusions.

Table 4: Stability analysis on FinHopper. Scores are reported as mean ± standard deviation in %.

Model	Mic Avg.	Mac Avg.
Human Performance	81.7 ± 1.6	78.7 ± 1.4
Claude-4.5-Sonnet-thinking	67.3 ± 0.8	62.1 ± 1.1
Claude-4.5-Sonnet-no-thinking	63.7 ± 1.0	58.1 ± 0.9
Gemini-3-Pro-thinking	65.0 ± 1.2	59.6 ± 0.9
Gemini-3-Pro-no-thinking	62.7 ± 1.1	55.7 ± 1.3
GPT-5.2-thinking	61.3 ± 0.7	54.6 ± 1.3
GPT-5.2	58.5 ± 1.5	51.3 ± 1.2
Doubao-seed-1-8-251215-thinking	61.1 ± 1.0	56.6 ± 0.8
Doubao-seed-1-8-251215	59.2 ± 1.3	53.6 ± 1.1
DeepSeek-V3.2-thinking	58.1 ± 1.4	46.5 ± 0.8
DeepSeek-V3.2-no-thinking	57.1 ± 1.7	45.4 ± 1.5
Kimi-k2-thinking	57.5 ± 0.9	54.4 ± 1.1
Kimi-k2	54.6 ± 1.2	51.4 ± 0.7
HunYuan-2.0-thinking	57.3 ± 1.1	51.0 ± 1.3
HunYuan-2.0-instruct	52.7 ± 1.6	46.0 ± 1.2
Grok-4.1-thinking	52.7 ± 0.8	48.1 ± 1.0
Grok-4.1	42.7 ± 1.4	42.7 ± 1.8
Qwen-max-20250923-thinking	52.5 ± 0.9	49.1 ± 1.0
Qwen-max-20250923-no-thinking	48.3 ± 1.3	45.2 ± 0.9
GLM-4.7-thinking	49.0 ± 1.5	42.5 ± 1.6
GLM-4.7-no-thinking	48.8 ± 1.8	40.9 ± 1.4

D State Transition Example

Table 6 presents illustrative examples of valid state transitions within our framework. For each transition type, we demonstrate how the answer obtained at hop k (highlighted in red) functions as a key constraint or subject for formulating the subsequent question at hop $k+1$.

E Single-Hop Pattern Taxonomy

We present a detailed taxonomy of 21 single-hop QA generation patterns in Table 7, organized into three categories: Entity-Time Comparison, Event-Based, and Entity Relationship patterns.

F Guide For Chain-Merge Verification

We show the detailed guide for Chain-Merge Verification in Table 8. For questions, criteria such as Intent Preservation and Logical Consistency maintain semantic integrity. They aim to eliminate ambiguity and prevent information leakage. For answers, criteria like Evidence Restriction and Faithfulness constrain outputs to the merged query. This ensures every intermediate answer is explicit and prohibits unsupported inferences. We provide concrete examples for each criterion to facilitate both automated and human evaluation.

F.1 Case Study

To investigate the underlying mechanisms of agentic reasoning, we conducted a qualitative analysis on the reasoning chains of all models (by thinking mode). Specifically, we sampled 20% of successful instances ($s_{final} = 1$) and 20% of failed instances ($s_{final} = 0$) from each difficulty level. This section summarizes the critical factors for success and categorizes the primary patterns of failure.

F.1.1 Mechanisms of Success

Our analysis reveals that solving multi-hop questions—particularly those exceeding 3 hops—requires a synergy of three core capabilities: strategic planning and decomposition, efficient tool utilization, robust context retention and self-correction.

As illustrated in Box 1 and Box 2, successful agents do not merely retrieve facts; they construct a logical roadmap. They maintain critical constraints across long horizons and autonomously trigger reflection mechanisms to correct deviations when intermediate steps yield implausible results.

F.1.2 Taxonomy of Failures

Despite possessing the capabilities mentioned above, agent performance degrades significantly as task complexity increases. By annotating and classifying the failed samples, we identified four primary failure modes within the reasoning chain.

1. **Decomposition and Planning Failures (38%):** This is the most prevalent structural error. Agents often exhibit logical sequencing errors, where the execution order of sub-goals is irrational, blocking subsequent steps. Furthermore, agents often lack the ability to refine their plans, leading to "dead ends" where they repetitively execute ineffective steps without strategic adjustment.

2. **Tool Misuse (31%):** This category involves the selection of inappropriate tools for the immediate sub-task. For instance, an agent might query a stock price tool instead of a corporate profile tool to find an IPO date. Additionally, failures occur due to improper usage, such as providing incorrect parameters or formats,

Table 5: Data pool selection based on input state.

Input State	**Data Pool**	**Examples**
Entity	Stocks and funds	Tencent, CATL, fund/ETF tickers
Time	Trading days, a period of time	2025-11-19; FY2025_Q3, 2024-10-05–2024-11-05
Sector	Industry classifications, sector indices	New Energy Vehicles; Semiconductor; CSI 300 sector index
Event	Earnings releases, dividends, corporate announcements, ...	Q3 earnings release; dividend distribution; share buyback announcement

Table 6: State transition rules with examples. Each row shows a valid transition illustrated with consecutive hop questions. The "Indicator" state terminates the reasoning chain.

From State	**To State**	**Hop k Question**	**Hop k+1 Question**
Entity	Entity	Question: Which fund holds the highest value of Moutai? *(Answer Type: Entity)*	Who is the fund manager of that fund? *(Answer Type: Entity)*
	Sector	Among 4 companies, which had the highest net profit margin in Q3 2025? *(Answer Type: Entity)*	What sector does that company belong to? *(Answer Type: Sector)*
	Time	Among 4 companies, which had the highest net profit margin in Q3 2025? *(Answer Type: Entity)*	When was that company listed? *(Answer Type: Time)*
	Indicator	What was CATL's closing price on Dec 30, 2025? *(Answer Type: Indicator)*	*369.20 CNY (Terminal, Answer Type: Indicator)*
Time	Entity	When did Jingjin Equipment Inc release its 2nd report in 2024? *(Answer Type: Time)*	On that day, which fund had weekly return $>$ -0.5%? *(Answer Type: Entity)*
	Indicator	Which day did Moutai's price peak in Nov 2025? *(Answer Type: Time)*	What was CATL's closing price on that day? *(Answer Type: Indicator)*
Sector	Entity	Which sector does CATL belong to? *(Answer Type: Sector)*	Which stock in that sector had highest 5-day gain? *(Answer Type: Entity)*
Event	Time	Kweichow Moutai dividend distribution *(Answer Type: Event)*	What was the date of that announcement? *(Answer Type: Time)*
Indicator	*(Terminal)*	*Numerical values serve as final answers and cannot constrain subsequent queries.*	

Table 7: Single-Hop QA Generation Patterns: Question Input → Answer Type.

#	Question	Condition	Answer	Example
Entity-Time Comparison Patterns				
1	ME + ST	Rank by indicator, select top-*n*	SE	Which of Tencent, Xiaomi, Alibaba, CATL had highest volume on 2025-11-19?
2	ME + ST	Multi-indicator filtering (unique match)	SE	Which of Tencent, Xiaomi, Alibaba, CATL had volume $>$ 100M and turnover $<$ 1% on 2025-11-19?
3	ME + ST	Compare quantitative indicator values	IND	What are net capital inflows for Tencent, Xiaomi, Alibaba on 2025-11-19? Which is largest?
4	SE + MT	Rank by indicator, select top-*n*	ST	On which day did Tencent have highest volume among 2025-11-17/18/19?
5	SE + MT	Compare quantitative indicator values	IND	What was Tencent's volume on 2025-11-17/18/19? Which day was highest?
6	SE + MT	First time crossing special threshold	ST	When did Apple's stock first exceed $150 among 2024-01/02/03?
7	SE + MT	Multi-indicator filtering (unique match)	ST	On which day among 2025-11-17/18/19 did Tencent have volume $>$ 100M and turnover $<$ 1%?
8	SE + MT	Query financial report release date	ST	When was Tencent's report released among Q1/Q2/Q3 2023?
9	ME + ST	Rank by indicator, select top-*n*	ME	Which two of Tencent, Xiaomi, Alibaba, CATL had highest volume on 2025-11-19?
10	SE + MT	Rank by indicator, select top-*n*	MT	Which two days of 2025-11-17/18/19 had highest volume for Tencent?
11	ME + MT	Rank times by cross-entity percentile mean	MT	Among 2025-11-17/18/19, which two days had best average volume for Tencent, Xiaomi, Alibaba?
12	ME + MT	Rank entities by cross-time percentile mean	ME	Among Tencent, Xiaomi, Alibaba, which two had best average volume across 2025-11-17/18/19?
13	ME + MT	Filter by trend (growth/decline)	ME	Which of Tencent, Xiaomi, Alibaba showed continuous growth across 2025-11-17/18/19?
Event-Based Patterns				
14	E + SE + NT	Extract time-related info	ST	When was Tencent founded? / When did BYD go public?
15	E + SE + NT	Query specific event date	ST	When was Tencent added to military list?
16	E + SE + NT	Locate by ordinal (1st/last/*n*-th)	ST	When was Tencent's first report release in 2024?
17	E + SE + NT	Locate multiple events (first *n*/all)	MT	What are all report release dates for Tencent in 2024?
18	E + NE + NT	Filter snippets, extract event date	ST	When did the Microsoft–Activision acquisition happen?
Entity Relationship Patterns				
19	SS	Query stock's sector/industry	SEC	What sector does BYD belong to?
20	SEC	Query sector constituent stocks	MS	What stocks are in the NEV sector?
21	SEC	Query sector constituent funds/ETFs	MF	What ETFs are related to the NEV sector?

ME=Multi-Entity, SE=Single-Entity, MT=Multi-Time, ST=Single-Time, E=Event, NE=No-Entity,
NT=No-Time, IND=Indicator, SS=Single-Stock, SEC=Sector, MS=Multi-Stock, MF=Multi-Fund

Table 8: Guide For Chain-Merge Verification.

Question	
Process Concealment	**Description**: Intermediate results and final answers must not be revealed via direct mentions or contextual cues. **Non-compliant Q**: "What was Apple's stock price on Dec 1, 2025, given it reported higher net income than NVIDIA in FY2026_Q3?" **Compliant Q**: "What was the stock price on Dec 1, 2025, of the company (Apple or NVIDIA) with higher net income in FY2026_Q3?"
Intent Preservation	**Description**: The semantic intent and information of both source questions must be preserved in the merged question. **Source Q**: · "Which company had higher net income in FY26_Q3: Apple or Nvidia?" · "What was Apple's stock price on Dec 1, 2025?" **Non-compliant Q**: "Between Apple and Nvidia, what was the Dec 1, 2025 stock price of the company with the better performance in FY26_Q3?" **Compliant Q**: Between Apple and Nvidia, what was the Dec 1, 2025 stock price of the company with the higher net income in FY26_Q3?
Disambiguation	**Description**: Critical constituents in questions, including entities, times, and metrics, must be explicitly specified and unambiguous. **Non-compliant Q**: "What was Apple's income last quarter?" **Compliant Q**: "What was Apple's Net Income in FY26_Q1?"
Closure	**Description**: All entities and metrics must remain within the scope of the original questions, rejecting extensions. **Source Q**: ·"Which company had higher net income in FY26_Q3: Apple or Nvidia?" · "What was Apple's stock price on Dec 1, 2025?" **Non-compliant Q**: "What was the stock price on Dec 1, 2025, for the company with higher FY26_Q3 net income: Apple, Nvidia, or Tesla?" **Compliant Q**: "What was the stock price on Dec 1, 2025, for the company with higher FY26_Q3 net income: Apple or Nvidia?"
Logical Consistency	**Description**: Causal links and premises must be reasonable and coherent. **Non-compliant Q**: "How did the Federal Reserve's first rate hike in 2025 drive the S&P 500's rally in 2022?" **Compliant Q**: "How did Apple's stock perform following the Federal Reserve's first rate hike in 2025?"
Answer	
Evidence Restriction	**Description**: The final answer must derive strictly from the newly merged single-hop question. **Source Q**: ·"Which company had the largest market cap in the S&P 500 on Dec 29, 2025?" · "What was Nvidia's stock price change in November 2025?" **Non-compliant Q**: "Which company had the highest price in the S&P 500 on December 29, 2025?" **Compliant Q**: "What was the stock price change in November 2025 of the company that had the largest market cap in the S&P 500 on Dec 29, 2025?"
Completeness	**Description**: All intermediate reasoning steps must be explicitly stated and supported by the step-wise answers. **Merge Q**: "What was the stock price change in November 2025 of the company that had the largest market cap in the S&P 500 on Dec 29, 2025?" **Non-compliant A**: "- $25.49" **Compliant A**: "As of Dec 29, 2025, the company with the largest market cap in the S&P 500 is Nvidia (NVDA). For November 2025, Nvidia's stock price showed a significant decline: - $25.49"
Faithfulness	**Description**: Conclusions must be inferred solely from the provided evidence, avoiding hallucinated claims or unjustified refusals. **Merge Q**: "Which company had the largest market cap in S&P 500 on Dec 29, 2025, and what was its FY26_Q3 revenue YoY growth?" **Non-compliant A1**: "As of December 29, 2025, the company with the largest market cap in the S&P 500 is Nvidia (NVDA). For its FY26_Q3, Nvidia reported a revenue growth of 62% year-over-year (YoY). AI demand will sustain its steady growth in Q4." **Non-compliant A2**: "Insufficient information to provide an answer." **Compliant A**: "As of December 29, 2025, the company with the largest market cap in the S&P 500 is Nvidia (NVDA). For its FY26_Q3, Nvidia reported a revenue growth of 62% year-over-year (YoY). "

which results in execution errors or null returns.

3. **Hallucination Propagation (17%):** When facing information gaps, agents may fabricate plausible but incorrect answers. This is particularly detrimental in multi-hop reasoning, as a single hallucinated intermediate step invalidates the entire subsequent chain.

4. **Information Extraction Failures (14%):** When tool outputs contain noise or irrelevant facts, models may misidentify these distractions as key clues. This misinterpretation causes the intermediate reasoning steps to deviate from the correct path.

Box 1: Success Case - 2 hop

System: $REACT_SYSTEM_PROMPT

User: ...<user_input_context>
[Question time]: 2025-12-10
[System time]: 2025-12-10
[User query]: Among 冠盛股份，大华股份，诺德股份，恒生电子 and 星华新材, which one recorded the highest net cash flow from operating activities in its 2025 Q1 interim report? On what date was the 2023 annual report of this company released?
</user_input_context>...

Agent: **<plan>** 1. Historical Analysis: No prior conversation history. This is the first query to retrieve the net cash flow from operating activities of the five companies in their 2025 Q1 interim reports, and then look up the release date of the 2023 annual report of the company with the highest figure.
2. Time Calibration: Query/system time is December 10, 2025. Target data includes the 2025 Q1 interim report (covering January–March 2025) and the release date of the 2023 annual report.
3. Action Plan: First, query the net cash flow from operating activities of the five companies in 2025 Q1, compare the figures to identify the top-performing company, then check the announcement date of its 2023 annual report. **</plan>**
<keyword> 冠盛股份 大华股份 诺德股份 恒生电子 星华新材 **</keyword>**

User: **<entity>**[{'name': '冠盛股份', 'codeForTools': 'sh605088', 'entity_type': 'GP_CH'}, {'name': '大华股份', 'codeForTools': 'sz002236', 'entity_type': 'GP_CH'}, {'name': '诺德股份', 'codeForTools': 'sh600110', 'entity_type': 'GP_CH'}, {'name': '恒生电子', 'codeForTools': 'sh600570', 'entity_type': 'GP_CH'}, {'name': '星华新材', 'codeForTools': 'sz301077', 'entity_type': 'GP_CH'}]**</entity>**

Agent: **<tool_call>**[{"id":"call_11614950","name":"get_cash_flow_statement_v2","arguments": {"end_date":"20250630","report_type":"Q1单季报","start_date":"20250101","stock_codes":"sh605088,sz002236,sh600110,sh600570,sz301077"}}]**</tool_call>**

Tool: **<tool_response>** [{"tool_call_id":"call_11614950","result":"Based on 大华股份 (Stock Code: 002236.SZ) ... and its Q1 2025 interim report with a statistical cutoff date of March 31, 2025, ... the net cash flow from operating activities stood at RMB -671,695,394.62, ...
Based on 星华新材 (Stock Code: 301077.SZ) ... and its Q1 2025 interim report with a statistical cutoff date of March 31, 2025, ... the net cash flow from operating activities stood at RMB 34,576,595.91, ...
Based on 诺德股份 (Stock Code: 600110.SH) and its Q1 2025 interim report with a statistical cutoff date of March 31, 2025, ... the net cash flow from operating activities stood at RMB 131,498,074.87, ...
Based on 恒生电子 (Stock Code: 600570.SH) and its Q1 2025 interim report with a statistical cutoff date of March 31, 2025, ... the net cash flow from operating activities stood at RMB -1,057,367,227.62, ...
Based on 冠盛股份 (Stock Code: 605088.SH) and its Q1 2025 interim report with a statistical cutoff date of March 31, 2025, ... the net cash flow from operating activities stood at RMB -137,607,237.29, ..."}] **</tool_response >**

Agent: **<plan>** 1. Historical Analysis: The tool returned the net cash flow from operating activities for Q1 2025: 诺德股份 recorded the highest figure at RMB 131,498,074.87, followed by 星华新材 at RMB 34,576,595.91; the figures for the rest were negative.
2. Time Calibration: The query was made on December 10, 2025. It is necessary to retrieve the 2023 annual report of 诺德股份 (the report covers the period ending December 31, 2023 and was released in 2024).
3. Action Plan: Look up the cash flow data in 诺德股份's 2023 annual report and extract its release date. **</plan>**
<keyword>诺德股份**</keyword>**

User: **<entity>**[{'name': '诺德股份', 'codeForTools': 'sh600110', 'entity_type': 'GP_CH'}]**</entity>**

Agent: **<tool_call>**[{"id":"call_44732059","name":"get_cash_flow_statement_v2","arguments": {"end_date":"20241231","report_type":"年报","start_date":"20240101","stock_codes":"sh600110"}}]**</tool_call>**

Tool: **<tool_response>** [{"tool_call_id":"call_44732059","result":"Based on the financial report data released by 诺德股份 (Stock Code: 600110.SH) on the A-share market on April 22, 2024, the annual report with a statistical cutoff date of December 31, 2024..."}] **</tool_response >**

Agent: **<plan>** 1. Historical Analysis: Net cash flow from operating activities for Q1 2025: 诺德股份 (sh600110) recorded the highest figure at RMB 131,498,074.87; however, the tool returned the 2024 annual report (for the period ending December 31, 2024, released on April 22, 2025) in the previous round, instead of the required 2023 annual report.
2. Time Calibration: The 2023 annual report (reporting period: December 31, 2023, released in 2024) is required. Adjust the query time range to January 1, 2023, to June 30, 2024.
3. Action Plan: Re-invoke the get_cash_flow_statement_v2 tool to retrieve the release date of 诺德股份's 2023 annual report.
</plan>
<keyword>诺德股份**</keyword>**

User: **<entity>**[{'name': '诺德股份', 'codeForTools': 'sh600110', 'entity_type': 'GP_CH'}]**</entity>**

Agent: **<tool_call>**[{"id":"call_42046231","name":"get_cash_flow_statement_v2","arguments": {"end_date":"20240630","report_type":"年报","start_date":"20230101","stock_codes":"sh600110"}}]**</tool_call>**

Tool: **<tool_response>** [{"tool_call_id":"call_42046231","result":"Based on the financial report data released by 诺德股份 (Stock Code: 600110.SH) on the A-share market on April 22, 2025, the annual report with a statistical cutoff date of December 31, 2023..."}]**</tool_response >**

Agent: **<answer>** Net Cash Flow from Operating Activities of the Five Companies (Unit: RMB)
冠盛股份 (sh605088): -13,760,723.72
大华股份 (sz002236): -671,695,394.62
诺德股份 (sh600110): 131,498,074.87 (Highest)
恒生电子 (sh600570): -1,057,367,227.62
星华新材 (sz301077): 34,576,595.91
The 2023 annual report of 诺德股份 (reporting period: December 31, 2023) was released on April 22, 2025. **</answer>**

Box 2: Success Case - 5 hop

System: $REACT_SYSTEM_PROMPT

User: ...<user_input_context>
[Question time]: 2025-12-10
[System time]: 2025-12-10
[User query]: What was the date of the third release of the Eurozone Consumer Price Index (CPI) in 2023? In the cumulative report of the quarter to which the release date belongs, between 国电电力 and 通宇通讯, which company recorded a higher year-on-year growth rate of net profit attributable to shareholders? What was the release date of the quarterly report corresponding to August 29, 2024 for that company? On that date, among the four funds: 广发纳指 100ETF 联接人民币 (QDII) A, 华宝纳斯达克精选股票 (QDII) C, 华宝中证有色金属 ETF 发起式联接 C and 嘉实沪深 300ETF, which two funds achieved the highest return rates over the past month? Between these two funds, which one had a larger fund size as of April 17, 2024?
</user_input_context>...

Agent: **<plan>** 1. Historical Analysis: This is the first planning exercise, with no historical dialogue data available.
2. Time Calibration: Both the query time and system time are January 4, 2026. It is necessary to look up the date of the third release of the Eurozone Consumer Price Index (CPI) in 2023, then query the cumulative quarterly report data of 国电电力 and 通宇通讯 for the quarter to which this date belongs, followed by the release date of the quarterly report corresponding to August 29, 2024, and finally the relevant data of the four funds as of April 17, 2024.
3. Action Plan:
Step 1: Look up the release dates of the Eurozone CPI in 2023 and determine the date of the third release.
Step 2: Based on the quarter to which this date belongs, query the year-on-year growth rate of net profit attributable to shareholders in the cumulative quarterly report of 国电电力 and 通宇通讯 for the corresponding quarter.
Step 3: Ascertain the release date of the quarterly report corresponding to August 29, 2024.
Step 4: Query the one-month return rate of the four funds and screen out the two with the highest returns.
Step 5: Query the fund size of the two selected funds as of April 17, 2024.**</plan>**
<keyword> Eurozone CPI **</keyword>**

User: **<entity>**[ ]**</entity>**

Agent:

<tool_call>[{"name": "get_global_macro", "params": {"names": "欧元CPI", "start_date": "2023-01-01", "end_date": "2023-12-31"}}]</tool_call>

Tool:

<tool_response> [{"tool_name":"get_global_macro","tool_params":{"names":"欧元CPI","start_date":"2023-01-01","end_date":"2023-12-31"},"result":"As of June 30, 2023: Eurozone core CPI (month-on-month): 0.4, Data release date: July 19, 2023 \n As of May 31, 2023: Eurozone core CPI (month-on-month): 0.3, Data release date: June 16, 2023 \nAs of April 30, 2023: Eurozone core CPI (month-on-month): 1.0 \nAs of March 31, 2023: Eurozone core CPI (month-on-month): 1.3 \nAs of February 28, 2023: Eurozone core CPI (month-on-month): 0.9; Data release date: March 17, 2023 \nAs of January 31, 2023: Eurozone core CPI (month-on-month): -0.4; Data release date: February 23, 2023"}]</tool_response >

Agent:

<plan> The third release date of the Eurozone CPI in 2023 was June 16, 2023, which covered the CPI data for May 2023. Next, we need to look up the year-on-year growth rates of net profit attributable to shareholders of the parent company of 国电电力 and 通宇通讯 as presented in their semi-annual reports (with statistics ending on June 30, 2023), which correspond to the date of June 16, 2023.</plan>
<keyword>国电电力 通宇通讯</keyword>

User:

<entity>[{'name': '国电电力', 'norm_code': '600795.SH', 'code': 'sh600795', 'entity_type': 'GP_CH'}, {'name': '通宇通讯', 'norm_code': '002792.SZ', 'code': 'sz002792', 'entity_type': 'GP_CH'}]</entity>

Agent:

<tool_call>[{"name": "get_financial_composite_indicator_v2", "params": {"stock_codes": "600795.SH,002792.SZ", "report_type": "Q3累计季报", "start_date": "20230101", "end_date": "20231231", "limit": 100}}]</tool_call>

Tool:

<tool_response> [{"tool_name":"get_financial_composite_indicator_v2","tool_params":{"stock_codes":"600795.SH,002792.SZ","report_type":"半年报","start_date":"20230101","end_date":"20231231","limit":100},"result":"Based on the financial report data of 通宇通讯 (Stock Code: 002792.SZ) released on August 23, 2023, in the A-share market, the key financial indicators of its semi-annual report (with a statistical cutoff date of June 30, 2023) are as follows (Monetary Unit: RMB Yuan):...\nYear-on-year growth rate of net profit attributable to shareholders: -16.79%;\nYear-on-year growth rate of net profit: -16.79%;\n3-year compound annual growth rate (CAGR) of operating income: -10.30%;\n3-year compound annual growth rate (CAGR) of net profit attributable to shareholders: 15.55%.\nBased on the financial report data of 国电电力 (Stock Code: 600795.SH) released on August 31, 2023, in the A-share market, the key financial indicators of its semi-annual report (with a statistical cutoff date of June 30, 2023) are as follows (Monetary Unit: RMB Yuan):...\nYear-on-year growth rate of net profit attributable to shareholders: 28.21%;\nYear-on-year growth rate of net profit: 16.72%;\n3-year compound annual growth rate (CAGR) of operating income: 18.89%;\n3-year compound annual growth rate (CAGR) of net profit attributable to shareholders: 63.57%."}]</tool_response >

Agent:

<plan> In the semi - annual report corresponding to June 16, 2023 (with the statistical cutoff date of June 30, 2023), the year - on - year growth rate of net profit attributable to shareholders of 通宇通讯 (002792.SZ) was - 16.79%, and that of 国电电力 (600795.SH) was 28.21%.\nThus, the year - on - year growth rate of net profit attributable to shareholders of 国电电力 was higher.\nNext, inquire about the release date of the cumulative quarterly report for the first three quarters (Q3 cumulative quarterly report) of 国电电力 corresponding to August 29, 2024.</plan>
<keyword>国电电力</keyword>

User:

<entity>[{'name': '国电电力', 'norm_code': '600795.SH', 'code': 'sh600795', 'entity_type': 'GP_CH'}]</entity>

Agent:

<tool_call>[{"name": "get_income_statement_v2", "params": {"stock_codes": "600795.SH", "report_type": "Q3累计季报", "start_date": "20230101", "end_date": "20231231", "limit": 100}}]</tool_call>

Tool:

<tool_response> [{"tool_name":"get_income_statement_v2","tool_params":{"stock_codes":"600795.SH","report_type":"Q3累计季报","start_date":"20230101","end_date":"20231231","limit":100},"result":"Based on the financial report data of 国电电力 (Stock Code: 600795.SH) released on October 25, 2024, in the A-share market, this refers to its revised cumulative Q3 quarterly report with a statistical cutoff date of September 30, 2023..."}]</tool_response >

Agent:

<plan> The release date of the cumulative Q3 quarterly report corresponding to August 29, 2024, for 国电电力 (600795.SH) was October 25, 2024. Next, look up the 1-month return rates of the following four funds as of October 25, 2024: 广发纳指 100ETF 联接人民币 (QDII) A, 华宝纳斯达克精选股票 (QDII) C, 华宝中证有色金属 ETF 发起式联接 C, and 嘉实沪深 300ETF; then identify the two funds with the highest 1-month return rates. </plan>
<keyword>广发纳指 100ETF 联接人民币 (QDII) A, 华宝纳斯达克精选股票 (QDII) C, 华宝中证有色金属 ETF 发起式联接 C, 嘉实沪深 300ETF</keyword>

User:

<entity>[{'name': '广发纳指100ETF联接人民币(QDII)A', 'norm_code': '021778.JJ', 'code': '021778.OF', 'entity_type': 'JJ'}, {'name': '华宝纳斯达克精选股票(QDII)C', 'norm_code': '017437.JJ', 'code': '017437.OF', 'entity_type': 'JJ'}, {'name': '华宝中证有色金属ETF发起式联接C', 'norm_code': '017141.JJ', 'code': '017141.OF', 'entity_type': 'JJ'}, {'name': '嘉实沪深300ETF联接(LOF)I', 'norm_code': '021886.JJ', 'code': '021886.OF', 'entity_type': 'JJ'}]</entity>

Agent:

<tool_call>[{"name": "get_fund_nav_period_return", "params": {"fund_codes": "021778.OF,017437.OF,017141.OF,017141.OF", "start_date": "20241025", "end_date": "20241025", "limit": 100}}]</tool_call>

Tool:

<tool_response> [{"tool_name":"get_fund_nav_period_return","tool_params": {"fund_codes":"021778.OF,017437.OF,017141.OF,017141.OF","start_date":"20241025","end_date":"20241025","limit":100}, "result":"...As of October 25, 2024, the 1-month return rates of the funds were as follows: 广发纳指 100ETF 联接人民币 (QDII) A at 1.656413%, 华宝纳斯达克精选股票 (QDII) C at 4.794021%, 华宝中证有色金属 ETF 发起式联接 C at 11.573175%, and 嘉实沪深 300ETF at 15.479115%..."}]</tool_response >

Agent:

<plan> As of October 25, 2024, the 1-month return rates of the funds were as follows: 广发纳指 100ETF 联接人民币 (QDII) A at 1.656413%, 华宝纳斯达克精选股票 (QDII) C at 4.794021%, 华宝中证有色金属 ETF 发起式联接 C at 11.573175%, and 嘉实沪深 300ETF at 15.479115%. Based on quantile comparison (a smaller quantile indicates a higher ranking), 嘉实沪深 300ETF had a quantile of 0% (ranked 1st), 华宝中证有色金属 ETF 发起式联接 C had a quantile of 33.33% (ranked 2nd), 华宝纳斯达克精选股票 (QDII) C had a quantile of 66.67% (ranked 3rd), and 广发纳指 100ETF 联接人民币 (QDII) A had a quantile of 100% (ranked 4th). Therefore, the top two funds with the highest 1-month return rates were 嘉实沪深 300ETF and 华宝中证有色金属 ETF 发起式联接 C. Next, look up the fund sizes of 嘉实沪深 300ETF and 华宝中证有色金属 ETF 发起式联接 C as of April 17, 2024. </plan>
<keyword>嘉实沪深 300ETF 华宝中证有色金属 ETF 发起式联接 C </keyword>

User:

<entity>[{'name': '华宝中证有色金属ETF发起式联接C', 'norm_code': '017141.JJ', 'code': '017141.OF', 'entity_type': 'JJ'}, {'name': '嘉实沪深300ETF联接(LOF)I', 'norm_code': '021886.JJ', 'code': '021886.OF', 'entity_type': 'JJ'}]</entity>

Agent:

<tool_call>[{"name":"get_fund_basic_info","params": {"fund_codes":"017141.OF,021886.OF","start_date":"20240417","end_date":"20240417"}}]</tool_call>

Tool:

<tool_response> [{"tool_name":"get_fund_basic_info","tool_params": {"fund_codes":"017141.OF,021886.OF","start_date":"20240417","end_date":"20240417"},"result":"...As of April 17, 2024, the fund sizes were as follows: 嘉实沪深 300ETF (Fund Code: 160724.OF) at RMB 276,053,239.26, and 华宝中证有色金属 ETF 发起式联接 C (Fund Code: 017141.OF) at RMB 42,204,365.14..."}]</tool_response >

Agent:

<answer> The third release date of the Eurozone CPI in 2023 was June 16, 2023, covering the CPI data for May 2023. In the semi-annual report corresponding to June 16, 2023 (with a statistical cutoff date of June 30, 2023), the year-on-year growth rate of net profit attributable to shareholders of 通宇通讯 (002792.SZ) stood at -16.79%, while that of 国电电力 (600795.SH) reached 28.21%. Thus, 国电电力 had a higher year-on-year growth rate of net profit attributable to shareholders.
The release date of the cumulative Q3 quarterly report corresponding to August 29, 2024 was October 25, 2024. As of October 25, 2024, the 1-month return rates of the funds were as follows: 广发纳指 100ETF 联接人民币 (QDII) A at 1.656413%, 华宝纳斯达克精选股票 (QDII) C at 4.794021%, 华宝中证有色金属 ETF 发起式联接 C at 11.573175%, and 嘉实沪深 300ETF at 15.479115%. Based on quantile comparison (a smaller quantile indicates a higher ranking), 嘉实沪深 300ETF had a quantile of 0% (ranked 1st), 华宝中证有色金属 ETF 发起式联接 C a quantile of 33.33% (ranked 2nd), 华宝纳斯达克精选股票 (QDII) C a quantile of 66.67% (ranked 3rd), and 广发纳指 100ETF 联接人民币 (QDII) A a quantile of 100% (ranked 4th). Therefore, the top two funds with the highest 1-month return rates were 嘉实沪深 300ETF and 华宝中证有色金属 ETF 发起式联接 C. As of April 17, 2024, the fund size of 嘉实沪深 300ETF (Fund Code: 160724.OF) was RMB 276,053,239.26, and that of 华宝中证有色金属 ETF 发起式联接 C (Fund Code: 017141.OF) was RMB 42,204,365.14. Hence, 嘉实沪深 300ETF had a larger fund size. </answer>

G Prompt of Single-Hop Pattern Taxonomy

Pattern 1: ME+ST → SE (Rank by indicator, select top-n)

[System Prompt]

Task Goal. Based on **entity list**, **time**, **tool names**, **tool descriptions**, and **tool retrieval results**, compare indicators across entities at the same time. Rank entities by a chosen indicator (higher/lower-is-better) and select the top-n. Question = **multi-entity + single-time**, answer = **single entity**.

Requirements: Question must mention all entities. Answer derived from tool results only. Compare percentile levels.

Failure: Output {} if 0 entities or invalid.

Output: `{"question":"...", "answer":"...", "entities":[<single>]}`

[User Prompt]

Entity List: `{entities}`
Time: `{time}`
Tool Name: `{tool_name}`
Tool Description: `{tool_descriptions}`
Tool Results: `{tool_result}`

Pattern 2: ME+ST → SE (Multi-indicator filtering, unique match)

[System Prompt]

Task Goal. Based on inputs, select n indicators. Find the **unique entity** satisfying a multi-condition filter (e.g., indicator A $>$ X AND indicator B $<$ Y). Question = **multi-entity + single-time**, answer = **unique entity** satisfying all conditions.

Requirements: Question includes all entities and time. Exactly one entity satisfies conditions. Answer lists all indicator values.

Failure: Output {} if no unique match.

Output: `{"question":"...", "answer":"...", "entities":[<single>]}`

[User Prompt]

Entity List: `{entities}`
Time: `{time}`
Tool Name: `{tool_name}`
Tool Description: `{tool_descriptions}`
Tool Results: `{tool_result}`

Pattern 3: ME+ST → IND (Compare quantitative indicator values)

[System Prompt]

Task Goal. Select a **quantitative numeric indicator** (volume, return, turnover). Construct QA: question = **multi-entity + single-time**, answer focuses on **indicator values themselves** (max/min, top-N values), not just "which entity".

Requirements: Question names all entities and time. Answer provides specific numeric values. `entities` field can be empty.

Failure: Output {} if invalid comparison.

Output: `{"question":"...", "answer":"...", "entities":[]}`

[User Prompt]

Entity List: `{entities}`
Time: `{time}`
Tool Name: `{tool_name}`
Tool Description: `{tool_descriptions}`
Tool Results: `{tool_result}`

Pattern 4: SE+MT → ST (Rank by indicator, select top-n)

[System Prompt]

Task Goal. Based on **entity**, **time list**, and **tool results**, compare a chosen indicator across different time points for the same entity. Rank times by indicator and select top-n. Question = **single-entity + multi-time**, answer = **single time**.

Requirements: Question mentions entity and all times. Answer details indicator values at each time.

Failure: Output {} if invalid question.

Output: `{"question":"...", "answer":"...", "entities":[<time>]}`

[User Prompt]

Entity List: `{entities}`
Time: `{time}`
Tool Name: `{tool_name}`
Tool Description: `{tool_descriptions}`
Tool Results: `{tool_result}`

Pattern 5: SE+MT → IND (Compare quantitative indicator values)

[System Prompt]

Task Goal. Select a quantitative indicator. Ask for **indicator values at each time point** and provide comparison. Question = **single-entity + multi-time**, answer emphasizes **indicator values and comparison result**.

Requirements: Question includes entity and all times. Answer gives specific numeric values for each time.

Failure: Output {} if invalid question.

Output: `{"question":"...", "answer":"...", "entities":[]}`

[User Prompt]

Entity List: `{entities}`
Time: `{time}`
Tool Name: `{tool_name}`
Tool Description: `{tool_descriptions}`
Tool Results: `{tool_result}`

Pattern 6: SE+MT → ST (First time crossing threshold)

[System Prompt]

Task Goal. Choose a culturally meaningful threshold (e.g., 66, 88, 100, 888) and ask for the **first time** the indicator **crosses above/below** that threshold. Question = **single-entity + multi-time**, answer = **single time**.

Requirements: Threshold should be meaningful. Ensure answer exists in data.

Failure: Output {} if no crossing found.

Output: `{"question":"...", "answer":"...", "entities":[<time>]}`

[User Prompt]

Entity List: `{entities}`
Time: `{time}`
Tool Name: `{tool_name}`
Tool Description: `{tool_descriptions}`
Tool Results: `{tool_result}`

Pattern 7: SE+MT → ST (Multi-indicator filtering, unique match)

[System Prompt]

Task Goal. Select n suitable indicators from tool description and results. Find the **unique time** when indicator A $>$ threshold AND indicator B $<$ threshold. Question = **single-entity + multi-time**, answer = **unique time** satisfying all conditions.

Requirements: Exactly one time satisfies conditions. Answer lists all indicator values at each time.

Failure: Output {} if no unique match.

Output: `{"question":"...", "answer":"...", "entities":[<time>]}`

[User Prompt]

Entity List: `{entities}`
Time: `{time}`
Tool Name: `{tool_name}`
Tool Description: `{tool_descriptions}`
Tool Results: `{tool_result}`

Pattern 8: SE+MT → ST (Query financial report release date)

[System Prompt]

Task Goal. From **tool retrieval results**, find the financial report release date. Construct a question asking for **the release date of a specific quarterly report**. Question = **single-entity + multi-time**, answer = **single time (release date)**.

Requirements: Question must specify which quarterly report. Answer includes exact release date.

Failure: Output {} if no release date found.

Output: `{"question":"...", "answer":"...", "entities":[<time>]}`

[User Prompt]

Entity List: `{entities}`
Time: `{time}`
Tool Name: `{tool_name}`
Tool Description: `{tool_descriptions}`
Tool Results: `{tool_result}`

Pattern 9: ME+ST → ME (Rank by indicator, select top-n)

[System Prompt]

Task Goal. Compare indicators across entities at the same time. Rank entities by a chosen indicator and select the **top-n entities** (where $n \geq 2$). Question = **multi-entity + single-time**, answer = **multiple entities**.

Requirements: Question mentions all entities. Answer compares percentile levels and returns ordered list.

Failure: Output {} if 0 entities or invalid.

Output: `{"question":"...", "answer":"...", "entities":[<list>]}`

[User Prompt]

Entity List: `{entities}`
Time: `{time}`
Tool Name: `{tool_name}`
Tool Description: `{tool_descriptions}`
Tool Results: `{tool_result}`

Pattern 10: SE+MT → MT (Rank by indicator, select top-n)

[System Prompt]

Task Goal. Compare a chosen indicator across different time points for the same entity. Rank time points by indicator and select **top-n times** (where $n \geq 2$). Question = **single-entity + multi-time**, answer = **multiple times**.

Requirements: Question mentions entity and all times. Answer returns ordered time list.

Failure: Output {} if invalid question.

Output: `{"question":"...", "answer":"...", "entities":[<times>]}`

[User Prompt]

Entity List: `{entities}`
Time: `{time}`
Tool Name: `{tool_name}`
Tool Description: `{tool_descriptions}`
Tool Results: `{tool_result}`

Pattern 11: ME+MT → MT (Rank times by cross-entity percentile mean)

[System Prompt]

Task Goal. For each time, compute the indicator's percentile mean (or equal-weighted average) across all entities as that day's composite score. Rank times and select top-n. Question = **multi-entity + multi-time**, answer = **multiple times**.

Requirements: Question includes all entities and times. Answer details scores and returns ranked times.

Failure: Output {} if invalid.

Output: `{"question":"...", "answer":"...", "entities":[<times>]}`

[User Prompt]

Entity List: `{entities}`
Time: `{time}`
Tool Name: `{tool_name}`
Tool Description: `{tool_descriptions}`
Tool Results: `{tool_result}`

Pattern 12: ME+MT → ME (Rank entities by cross-time percentile mean)

[System Prompt]

Task Goal. For each entity, compute the indicator's percentile mean (or equal-weighted average) across all given times as that entity's composite score. Rank entities and select top-n. Question = **multi-entity + multi-time**, answer = **multiple entities**.

Requirements: Question includes all entities and times. Answer details scores and returns ranked entities.

Failure: Output {} if invalid.

Output: `{"question":"...", "answer":"...", "entities":[<list>]}`

[User Prompt]

Entity List: `{entities}`
Time: `{time}`
Tool Name: `{tool_name}`
Tool Description: `{tool_descriptions}`
Tool Results: `{tool_result}`

Pattern 13: ME+MT → ME (Filter by trend: growth/decline)

[System Prompt]

Task Goal. Select an indicator and determine each entity's **trend** over the given time period (e.g., continuous growth, continuous decline, rise-then-fall, etc.). Find entities matching a specific trend. Question = **multi-entity + multi-time**, answer = **multiple entities** matching the trend.

Requirements: Question asks "which entities show [trend]?" Answer lists indicator values across times.

Failure: Output {} if no entities match.

Output: `{"question":"...", "answer":"...", "entities":[<list>]}`

[User Prompt]

Entity List: `{entities}`
Time: `{time}`
Tool Name: `{tool_name}`
Tool Description: `{tool_descriptions}`
Tool Results: `{tool_result}`

Pattern 14: E+SE+NT → ST (Extract time-related info)

[System Prompt]

Task Goal. Based on **entity** and **tool results**, extract time-related information (founding date, listing date, etc.). Construct a QA pair: question = **single entity + no explicit time input**, answer = **single time**.

Requirements: Answer must be exact date from tool results. No fabrication.

Failure: Output {} if no time info available.

Output: `{"question":"...", "answer":"...", "entities":[<time>]}`

[User Prompt]

Entity List: `{entities}`
Tool Name: `{tool_name}`
Tool Description: `{tool_descriptions}`
Tool Results: `{tool_result}`

Pattern 15: E+SE+NT → ST (Query specific event date)

[System Prompt]

Task Goal. From **tool results**, find dates of specific events (regulatory events, company milestones, business events, personnel changes, etc.). Construct a question asking for a **specific event date**. Question = **single entity**, answer = **single time**.

Requirements: For periodic events, specify the year. Answer must be exact date.

Failure: Output {} if no event date found.

Output: `{"question":"...", "answer":"...", "entities":[<time>]}`

[User Prompt]

Entity List: `{entities}`
Tool Name: `{tool_name}`
Tool Description: `{tool_descriptions}`
Tool Results: `{tool_result}`

Pattern 16: E+SE+NT → ST (Locate by ordinal: 1st/last/n-th)

[System Prompt]

Task Goal. From **tool results**, find periodic event dates (report releases, dividends, shareholder meetings, etc.). Construct a question using **ordinal keywords** (first, last, n-th) to locate a specific event date. Question = **single entity**, answer = **single time**.

Requirements: Must use ordinal keywords. Need $\geq$2 same-type events for ranking.

Failure: Output {} if $<$2 events found.

Output: `{"question":"...", "answer":"...", "entities":[<time>]}`

[User Prompt]

Entity List: `{entities}`
Tool Name: `{tool_name}`
Tool Description: `{tool_descriptions}`
Tool Results: `{tool_result}`
Current Time: `{current_time}`

Pattern 17: E+SE+NT → MT (Locate multiple events: first n/all)

[System Prompt]

Task Goal. From **tool results**, find multiple periodic event dates. Construct a question using keywords like "first n", "last n", or "all" to locate **multiple event dates**. Question = **single entity**, answer = **multiple times** ($\geq$2).

Requirements: Answer must contain $\geq$2 times. For current year, add cutoff date constraint.

Failure: Output {} if $<$2 events found.

Output: `{"question":"...", "answer":"...", "entities":[<times>]}`

[User Prompt]

Entity List: `{entities}`
Tool Name: `{tool_name}`
Tool Description: `{tool_descriptions}`
Tool Results: `{tool_result}`
Current Time: `{current_time}`

Pattern 18: E+NE+NT → ST (Filter snippets, extract event date)

[System Prompt]

Task Goal. Based on **event description** and **tool results**: (1) Filter relevant snippets matching the event description; (2) Extract the actual **event occurrence date** (not publish/report date); (3) Construct a question asking when the event occurred. Answer = **single time**.

Requirements: Event date must be exact, not news publish time.

Failure: Output {} if no actual event date found.

Output: `{"question":"...", "answer":"...", "entities":[<time>]}`

[User Prompt]

Event Description: `{event}`
Tool Name: `{tool_name}`
Tool Description: `{tool_descriptions}`
Tool Results: `{tool_result}`

Pattern 19: SS → SEC (Query stock's sector/industry)

[System Prompt]

Task Goal. Based on **entity list** (single stock/company) and **tool results**, generate a single-hop QA pair: "Stock → Sector/Industry". Question asks which sector/industry the stock belongs to. Answer = **single sector/industry name**.

Requirements: Answer must come from tool results. No fabrication.

Failure: Output {} if sector info unavailable.

Output: `{"question":"...", "answer":"...", "entities":[<sector>]}`

[User Prompt]

Entity List: `{entities}`
Time: `{time}`
Tool Name: `{tool_name}`
Tool Description: `{tool_descriptions}`
Tool Results: `{tool_result}`

Pattern 20: SEC → MS (Query sector constituent stocks)

[System Prompt]

Task Goal. Based on **entity list** (single sector/industry/concept) and **tool results**, generate a single-hop QA pair: "Sector → Constituent Stocks". Question asks which stocks belong to the sector. Answer = **multiple stock names**.

Requirements: Answer must come from tool results. `entities` is a list of stock names.

Failure: Output {} if constituent list unavailable.

Output: `{"question":"...", "answer":"...", "entities":[<stocks>]}`

[User Prompt]

Entity List: `{entities}`
Time: `{time}`
Tool Name: `{tool_name}`
Tool Description: `{tool_descriptions}`
Tool Results: `{tool_result}`

Pattern 21: SEC → MF (Query sector constituent funds/ETFs)

[System Prompt]

Task Goal. Based on **entity list** (single sector/industry/concept) and **tool results**, generate a single-hop QA pair: "Sector → Constituent Funds/ETFs". Question asks which funds/ETFs are related to the sector. Answer = **multiple fund/ETF names**.

Requirements: Answer must come from tool results. `entities` is a list of fund/ETF names.

Failure: Output `{}` if fund list unavailable.

Output: `{"question":"...", "answer":"...", "entities":[<funds>]}`

[User Prompt]

Entity List: `{entities}`
Time: `{time}`
Tool Name: `{tool_name}`
Tool Description: `{tool_descriptions}`
Tool Results: `{tool_result}`

Pattern Color Legend

ME+ST → SE | ME+ST → IND | SE+MT → ST/IND | ME+ST → ME
SE+MT → MT | ME+MT → MT/ME | Event-based | Entity Relationship

Abbreviations: ME=Multi-Entity, SE=Single-Entity, MT=Multi-Time, ST=Single-Time, E=Event, NE=No-Entity, NT=No-Time, IND=Indicator, SS=Single-Stock, SEC=Sector, MS=Multi-Stock, MF=Multi-Fund

H Prompt of React Framework

Prompt of React Framework

[System Prompt]
 # Role
You are a high-IQ financial decision-making agent. Your task is to decide whether to invoke tools to obtain new information or summarize the final answer using existing information in the next step, based on the context, to solve the user's core problem.
 # Special Instructions
1. [Mandatory Memory Application]: ...
2. [Multi-round Execution]: ...
3. [Tool List]: ...
4. [Mandatory Tool Execution]: ...
- CRITICAL RULE: When the system provides you with information in the format of $\langle entity\rangle ... \langle/entity\rangle$, it means you have obtained relevant entity information but no specific data yet. In this case, you must and only invoke tools (function call).
- Strictly prohibit outputting any analytical text.
- Strictly prohibit directly fabricating stock prices, financial report data, or answering questions solely based on entity codes.
- Example of Violation: After seeing "Jikai Co., Ltd.: sz002691", directly outputting "The current stock price of Jikai Co., Ltd. is..." (This is an absolutely forbidden hallucination).
 # Output Format (Requirements)
Please select the only corresponding output format below according to the current dialogue status:
Status A: Planning Required (Start / Loop)
If there is no $\langle entity\rangle$ information available currently and you need to query data: ...
(Stop output and wait for the system to return $\langle entity\rangle$ information)
Status B: Entities Ready (Tool Execution)
Trigger Condition: If and only if the input context contains the $\langle entity\rangle ... \langle/entity\rangle$ information returned by the system. In this status, you must invoke tools (function call) and are prohibited from outputting any other content.
Status C: Sufficient Information (Answer)
Trigger Condition: If and only if the data in historical dialogues completely covers all dimensions of the [User Input]. If the user's question involves multiple dimensions (e.g., multi-entity, multi-time comparison), you must obtain data for each dimension by invoking tools, without exception.
Important Notes
Unit Conversion Verification: In the results returned by tools, ...and strictly prohibit conclusion distortion caused by calculation or unit errors...

[User Prompt]

 # Context Data
User Input Context (Including Time Benchmark)
<user_input_context>
[Query Time]: {query_time}

[System Time]: {current_time}
[User Query]: {user_question}
</user_input_context>
Notes: - [Query Time]: Time entities (e.g., today, recently, etc.) are based on the query time. - [System Time]: Current system time.
 # Execution Now, based on the above information, output the next content in accordance with the **Output Format** specified in the System Prompt.

I Prompt of Evaluation

Prompt of Final Answer Evaluation

[System Prompt]
Evaluate the Student Answer objectively and rigorously against the provided Standard Answer. You must abandon any tendency to solve the problem independently and strictly act as an execution tool for evaluating answers.

Final Answer Scoring Rules (Evaluate Only the Final Answer, Ignore the Process)
You are only allowed to check whether the Final Answer is consistent with the Standard Answer (ignore format differences such as units and percentage conversions).
- If consistent: Award 1 point.
- If inconsistent: Award 0 point.
No scores between 0 and 1 are permitted; no points shall be added or deducted for the process.

You must provide feedback in the following format strictly, with no redundant opening remarks:
1. [Scoring Basis]: State whether the final answers are consistent in no more than 150 words.
2. [JSON]:

```
{"answer_score": score_value}
```

The grading task officially begins:

[User Prompt]

<Question>: {prompt}
<Standard Answer>: {response_reference}
<Student Answer>: {response}

Prompt of Process Evaluation

[System Prompt]
Evaluate the Student Answer objectively and rigorously against the provided Standard Answer. You must abandon any tendency to solve the problem independently and strictly act as an execution tool for evaluating answers.

Process Answer Scoring Rules (Evaluate Only the Process, Ignore the Final Answer Entirely)
The Standard Answer contains both "Process Answers" and "Final Answers" (or similar summaries/conclusions). You must:
- Extract and score only the process-related key points such as "process answers, intermediate steps, sub-question answers";
- Explicitly exclude final conclusion content such as "final answers, final conclusions, therefore, in summary", which shall not be counted into N nor affect the score.

Scoring Method (N Process Key Points, 0/1 Scoring for Each Item):
1) Decompose the question and the corresponding key points of the Standard Answer, and count the total number of process key points in the Standard Answer as N (excluding the final answer).
2) For each process key point (which is the answer to an intermediate sub-question rather than supporting evidence), judge whether the Student Answer provides the corresponding and correct process conclusion/intermediate result: mark 1 for correct, 0 for incorrect or missing.
3) Full score for the process part is 0.5 points: Total Score = 0.5 * (Number of Hit Process Key Points) / N. If N=0, the total score shall be 0.
Evaluation Criterion: The Standard Answer is absolutely correct and indisputable.
If the Standard Answer requires "multiple choices/multiple key points", the student must cover all of them. Missing any sub-item will result in 0 score for that sub-item.
Compatibility: Substantial equality of numerical values shall be deemed correct (e.g., 0.15 = 15%), and missing units shall not affect the correctness judgment.

You must provide feedback in the following format strictly, with no redundant opening remarks:
1. [Scoring Basis]: Briefly state the value of N, which process key points are hit and which are missed, within 150 words.
2. [JSON]:

```
{"answer_score": score_value}
```

Example:
Question
For the three companies Guqi Cashmere, Zhongda Leader, and Binhai Energy, in their Q3 single quarterly reports corresponding to October 16, 2024, which company had a net profit exceeding 10 million yuan and an income tax expense below 100,000 yuan? When was that company listed? On the listing day of that company, which fund between Harvest CSI Chuangke Chuangye 50 Index C and Tianhong Yongfeng Stable Pension Target 1-Year Holding Hybrid (FOF) had a 1-year return rate exceeding 10% and a 3-month return rate being negative?
Standard Answer
In the Q3 single quarterly reports corresponding to October 16, 2024, Binhai Energy had a net profit of 1.8317 million yuan and an income tax expense of 1.3599 million yuan; Guqi Cashmere had a net profit of 40.6356 million yuan and an income tax expense of 0 yuan; Zhongda Leader had a net profit of 13.3423 million yuan and an income tax expense of 0.3693 million yuan. Only Guqi Cashmere met the conditions of net profit exceeding 10 million yuan (40.6356 million yuan > 10 million yuan) and income tax expense below 100,000 yuan (0 yuan < 100,000 yuan). Guqi Cashmere was listed on the A-share market on May 29, 2025. On May 29, 2025, Harvest CSI Chuangke Chuangye 50 Index C had a 1-year return rate of 12.66% and a 3-month return rate of -9.01%; Tianhong Yongfeng Stable Pension Target 1-Year Holding Hybrid (FOF) had a 1-year return rate of 2.69% and a 3-month return rate of 0.86%. Only Harvest CSI Chuangke Chuangye 50 Index C met the conditions of 1-year return rate exceeding 10% and 3-month return rate being negative.
Student Answer
Only Guqi Cashmere met the conditions of net profit exceeding 10 million yuan (40.6356 million yuan > 10 million yuan) and income tax expense below 100,000 yuan (0 yuan < 100,000 yuan). Guqi Cashmere was listed on the A-share market on May 29, 2025. On May 29, 2025, only Harvest CSI Chuangke Chuangye 50 Index C met the conditions of 1-year return rate exceeding 10% and 3-month return rate being negative.
[Scoring Basis]
N is 2. The student answer hit the two process key points of Guqi Cashmere and May 29, 2025, with no omissions. 0.5 * 2 / 2 = 0.5
[JSON]:

```
{"answer_score": 0.5}
```

The grading task officially begins:

[User Prompt]

<Question>: {prompt}
<Standard Answer>: {response_reference}
<Student Answer>: {response}

J Prompt of Quality Control

J.1 Single-Hop Verification

Prompt of Single-Hop Verification

[System Prompt]
You are a professional answer comparison specialist, adept at determining whether the original answer can be directly derived from the reference materials.

[User Prompt]

Please strictly determine whether the original answer can be **directly derived** from the reference materials. Key information such as numerical values, percentages, dates and units **must be exactly consistent**.

Question: {query}
Original Answer: {original_answer}
Reference Materials: {content}

Please return **only a JSON object** in the following format:

```
{
  "can_derive": true/false,
  "reason": "Briefly state the reason"
}
```

Notes:
1. If the core information of the original answer can be directly obtained from the reference materials, return ‘true‘ even if the expression is different.
2. Return ‘false‘ if there are discrepancies in numerical values, dates or key information.
3. Return ‘false‘ if the question does not contain explicit time information.
4. Do not include any explanatory text; return only the JSON object.

J.2 Chain-Merge Verification

Prompt of Question Critic

[System Prompt]
 # Task Objective
Validate and, if necessary, optimize and rewrite the "draft merged question" to ensure that:
- The defined scope of entities and time periods is strictly retained and explicitly presented, without expanding to the entire market or ambiguous sets;
- No information in the original question is lost (including entities, time periods, indicators, comparison relationships, etc.);
- No answers are revealed in advance, and only the question is posed;
- The language is natural, uniquely solvable, and answerable using the provided data.

 # Usage Instructions
You will receive the following inputs:
- Original QA pair (`original_qa`)
- QA pair to be merged (`new_qa`)
- Draft question text appended after the phrase "Draft Merged Question:" at the bottom of the message

 # Notes
Core Principles
- The preceding question must not reveal the answer to the subsequent question, and vice versa.
- ...

Reference Specification
- Prohibit the use of references that cause scope expansion, e.g., "the company with the highest trading volume on 2025-11-10".
- ...

Information Completeness
- Do not add, delete, or weaken any existing entities.
- ...

Uniqueness and Solvability
- The question must be unique, solvable, and have a controllable search scope.
- ...

Expression Specification
- Avoid redundant sub-questions (content already described in the previous question does not need to be repeated).
- ...

 # Recommended Multi-step Splicing Structure
- In the first step, pose the question using an explicit set of entities.
- In the second step, use "Among the entity/time set from the previous question, the company/fund/market/day/quarter/year with... " to connect, and make comparisons with newly added entities or conditions.
- In subsequent steps, use "this company/that fund/that market/that day/that quarter/that year..." to build on the results of the previous step for further comparisons.
- Maintain a non-expanding scope, unambiguous expression, and no answer leakage throughout the process.

 # Correct Example

> Which company had the highest trading volume on 2025-11-10, Giant Network or Century Huatong?
> Between the company with higher trading volume among Giant Network and Century Huatong and YooZoo Network, which one had a larger stock price increase on November 12, 2025?
> Between this company and Industrial and Commercial Bank of China, Daqin Railway, and Tongyi Co., Ltd., which one had the highest turnover rate on 2025-11-18?

 # Counterexample Explanations
Counterexample 1: Answer Leakage (Directly revealing intermediate answers)
Incorrect: "...the company with higher transaction volume... (answer: Century Huatong) among [Century Huatong, Industrial and Commercial Bank of China...]..."
Correction: Use referential terms such as "that company" or "this company" for connection, without disclosing the specific entity name.

Counterexample 2: Time Answer Leakage
Incorrect: "Among the two days of 2025-05-22 and 2025-03-18 for Bubugao, on the day with higher trading volume... on May 22, 2025..."
Correction: Specific dates should not be mentioned in subsequent parts; use "the financial report quarter corresponding to that day" instead.

Counterexample 3: ...

 # Output Format
Return only the JSON object:

{

```
  "question": "<Optimized question or original question>"
}
```

[User Prompt]

Original QA Pair:
{original_qa}
QA Pair to Be Merged:
{new_qa}

Prompt of Answer Critic

[System Prompt]
 # Task Objective

Validate the *draft merged answer* to ensure that:
- All factual details from the original QA pair and the QA pair to be merged are fully retained without omission or tampering;
- The answer responds smoothly to the merged question with coherent logic;
- The `entities` field correctly points to the answer entities/time periods corresponding to the last sub-question.

 # Usage Instructions

You will receive the following inputs:
- Original QA pair (`original_qa`)
- QA pair to be merged (`new_qa`)
- Draft question text appended after the phrase "Draft Merged Question:" at the bottom of the message

 # Validation Rules

Completeness Validation
- All details (entities, numerical values, time periods, rankings, etc.) in the answer of the original QA pair must be fully retained;
- All details in the answer of the QA pair to be merged must be fully retained;
- No information from any sub-answer is allowed to be omitted.

Accuracy Validation
- All content must be directly derived from the original QA pair or the QA pair to be merged; fabrication or speculation is strictly prohibited;
- Numerical values, entity names, time periods, etc., must be consistent with the original data;
- Tampering with or misrepresenting the meaning of the original answers is prohibited.

 # Output Format

Return only the JSON object:

```
{
  "answer": "<Optimized answer or original answer>"
}
```

[User Prompt]

Original QA Pair:
{original_qa}
QA Pair to Be Merged:
{new_qa}