Journal of Science, Innovation & Social Impact

2026 2nd International Conference on Intelligent Computing and Automated Systems (ICAS 2026)

Article

An Empirical Comparison of ReAct, Reflexion, Plan-and-Solve, and Tree-of-Thought Planning Strategies on Financial Question Answering and Numerical Reasoning Tasks

Xuanyi Fu [1,*], Tianxing Tang [2] and Chuankai Luo [3]

[1] M.S.E. in Computer Science, Johns Hopkins University, Baltimore, MD, USA
[2] Translation and Localization Management, Middlebury Institute of International Studies, Monterey, CA, USA
[3] Department of Electronic Engineering, Tsinghua University, Beijing, China
* Correspondence: Xuanyi Fu, M.S.E. in Computer Science, Johns Hopkins University, Baltimore, MD, USA

Abstract: Large language model agents increasingly automate reasoning- and decision-intensive financial workflows, yet the comparative effectiveness of competing planning strategies on finance-specific tasks remains unclear. We conduct a controlled empirical comparison of four widely adopted planning strategies --- ReAct, Reflexion, Plan-and-Solve, and Tree-of-Thought --- on four public financial benchmarks spanning multi-step numerical reasoning (FinQA), multi-turn numerical dialogue (ConvFinQA), hybrid tabular-textual question answering (TAT-QA), and long-document question answering (DocFinQA). Using a shared GPT-4o backbone, a common tool set, and a unified evaluation protocol, we measure execution accuracy, exact-match correctness, per-task-type performance, and per-query token cost across three random seeds. Plan-and-Solve offers the best accuracy-per-dollar on purely numerical tasks, delivering a moderate 2.8-point improvement over ReAct on FinQA at roughly one-seventh the token budget of Tree-of-Thought. ReAct with retrieval dominates on long-document DocFinQA, outperforming Plan-and-Solve by 4.1 points. Tree-of-Thought attains the single highest accuracy on the compound-arithmetic subset of TAT-QA (71.4%) but costs 7.2× more tokens per query than Plan-and-Solve. A manual error typology across 400 failures confirms that each strategy repairs a distinct failure class, and that no single strategy dominates all four financial task types. The findings clarify an existing design-space question rather than propose new methodology.

Keywords: LLM agents; planning strategies; financial question answering; empirical evaluation

Received: 18 March 2026
Revised: 25 April 2026
Accepted: 09 May 2026
Published: 13 May 2026

Copyright: © 2026 by the authors. Submitted for possible open access publication under the terms and conditions of the Creative Commons Attribution (CC BY) license (https://creativecommons.org/licenses/by/4.0/).

1. Introduction

1.1. Motivation and Background

Large language models increasingly serve as the reasoning core of autonomous agents deployed in finance, where tasks range from extracting numerical facts from earnings reports to forecasting and executing sequential trading decisions. The planning strategy --- the policy by which an agent selects reasoning steps, invokes tools, and consolidates intermediate results --- has emerged as a critical design choice that often matters more than the underlying base model [1]. Four strategies have gained particular traction in recent literature: ReAct interleaves verbal reasoning with action calls; Reflexion adds a verbal self-correction loop; Plan-and-Solve splits each query into an explicit plan followed by step execution; and Tree-of-Thought expands candidate reasoning paths and prunes them with an internal evaluator. Finance is an especially demanding application

domain: numerical correctness must be exact, evidence is distributed across tables and prose, and disclosures routinely exceed one hundred pages [2].

1.2. Research Gap in Comparing Planning Strategies on Financial Tasks

The existing literature on planning strategies and on financial LLM agents has developed along two parallel tracks that rarely meet. General-purpose comparisons of planning strategies take place on benchmarks such as HotpotQA, GSM8K, and Game-of-24, while financial agent studies propose new role-play or memory architectures and report end-to-end accuracy rather than isolating the effect of the planning loop [3].

1.2.1. Fragmented Evidence in Existing Literature

AgentBench evaluates eight agent environments with ReAct and chain-of-thought baselines but excludes Reflexion and Tree-of-Thought, and covers no financial domain tasks [4]. Financial benchmarks such as FinBen enumerate tasks and score base LLMs without separating the contribution of the planning strategy [5]. A single published comparison cannot on its own answer whether a moderate accuracy gap between, say, Plan-and-Solve and ReAct on FinQA reflects a property of the dataset or a property of the planning loop, and this ambiguity propagates into downstream engineering decisions about which strategy to deploy in a given financial application.

1.2.2. Why Finance Is a Discriminating Test-Bed

Finance exercises three capabilities that planning strategies target unevenly. Numerical correctness favours strategies with explicit decomposition or search, because a single arithmetic slip invalidates the entire answer. Hybrid tabular-textual evidence favours strategies that can re-visit earlier reasoning steps, since extraction errors propagate silently into later computations [6]. Long-document contexts favour tool-augmented agents that retrieve selectively, since greedy single-pass reasoning cannot fit an entire 10-K filing into working memory. These three capabilities align one-to-one with Plan-and-Solve, Reflexion, and ReAct respectively, making finance a natural stress test for their differences [7].

1.3. Contributions

We present the first controlled head-to-head comparison of ReAct, Reflexion, Plan-and-Solve, and Tree-of-Thought on four public financial benchmarks under a shared backbone and tool set. The comparison contributes a unified evaluation protocol that combines execution accuracy, exact-match correctness, and a five-way error typology; a cost-accuracy trade-off analysis with per-query token counts; a per-task-type breakdown that isolates each strategy's contribution to compound arithmetic, single-step extraction, and long-document retrieval; and a public release of all prompts, tool wrappers, and evaluation scripts to enable exact reproduction of every reported number [8,9].

2. Related Work

2.1. LLM-Agent Planning Strategies

Planning strategies for LLM agents can be grouped by whether their outer loop is organised around reasoning steps or around environment actions. The former family traces its lineage to chain-of-thought prompting and has diverged into path-enumerating, decomposition-based, and search-based variants. The latter family is rooted in tool-use and closed-loop interaction with an environment [10].

2.1.1. Reasoning-Centric Strategies

Chain-of-thought prompting demonstrated that encouraging a model to verbalise intermediate steps improves multi-step numerical and symbolic reasoning. Plan-and-Solve reformulates this prompt into an explicit two-stage schedule in which the agent first devises a written plan and then executes each sub-step, which measurably reduces the rate of missed reasoning steps on arithmetic benchmarks [11]. Tree-of-Thought generalises single-path chain-of-thought into a tree where multiple candidate thoughts are expanded at each level and pruned by the LLM acting as its own evaluator; the

strategy achieves large accuracy gains on search-like puzzles at proportionally increased inference cost [12]. Self-consistency replaces greedy decoding with majority voting over sampled chain-of-thought paths; Least-to-Most decomposes a problem into sequentially dependent sub-problems; Graph-of-Thought further generalises the tree into a directed acyclic graph with aggregation and feedback edges [13]. The strategies differ in how broadly they explore the reasoning space and in how they re-combine or select among partial solutions.

2.1.2. Action-Centric Strategies

ReAct interleaves verbal reasoning with discrete actions such as tool calls or retrieval queries, allowing an agent to ground its reasoning in external observations. Reflexion adds an outer verbal-reinforcement loop in which a Self-Reflection module converts execution feedback into a lesson that is appended to episodic memory before the next attempt [14]. The action-centric family is naturally suited to tasks where evidence is external, retrieval is imperfect, or computation must be offloaded to a calculator.

2.2. LLM Agents and Evaluation in Finance

Architecture-level contributions in financial agents extend base models with role-play, tool use, and multimodal inputs; a recent tool-augmented multimodal trading agent couples price, news, and chart evidence with a diversified memory and reports state-of-the-art returns on six asset datasets [15]. These contributions are largely orthogonal to the planning-strategy question we study, since they vary the base model or the surrounding agent architecture while leaving the planning loop fixed to a single strategy. Evaluation in finance has followed a parallel trajectory [16]. FinBen consolidates dozens of datasets across 24 tasks spanning information extraction, textual analysis, question answering, forecasting, risk management, and a novel stock-trading agent task. Additional benchmarks target specific reasoning axes --- multi-step numerical reasoning, multi-turn dialogue, hybrid tabular-textual evidence, full-filing long context, and program-synthesis question answering --- and the four we use are introduced in detail in Section 3. Despite the density of the benchmark landscape, published scores are typically reported for one planning strategy (often ReAct or plain chain-of-thought) and one backbone, so the cross-strategy comparison that motivates the present study has not been performed at the scale and with the experimental controls that the four strategies require to be compared fairly [17,18]. The benchmarks we use here are the most commonly cited subset; their combined scale is sufficient to support statistical comparison while remaining feasible for the token budgets that Tree-of-Thought demands.

3. Experimental Setup

3.1. Task Formulation and Evaluation Protocol

Each evaluation instance is a tuple (question, evidence, tool set, budget) where the evidence is either a set of tables, a hybrid table-paragraph context, or a full SEC filing; the tool set includes a Python calculator, a table-row look-up, and an optional BM25 retriever; and the budget caps the number of reasoning or action steps at 15. An agent instantiates one of the four planning strategies, issues a sequence of steps that may include tool calls, and commits a final answer [19]. Two evaluation metrics are applied uniformly. Execution accuracy measures whether the answer --- after the generated reasoning program is executed --- matches the gold value within a relative tolerance of 0.5 percent, following the convention of program-of-thought evaluation. Exact-match measures whether the textual span returned by the agent matches the gold span for extraction questions. All reported numbers are averaged across three random seeds, and significance is assessed with a paired bootstrap test at the 95 percent confidence level [20,21]. The same prompt template, temperature, maximum step budget, and tool signatures are shared across all four strategies so that differences in accuracy are attributable to the planning loop rather than to prompt engineering.

3.2. Planning Strategies under Comparison

All four strategies are instantiated from the original prompts and hyperparameters published with their respective papers and are kept free from any task-specific prompt tuning beyond the shared scaffold.

3.2.1. Single-Trajectory Strategies: ReAct and Plan-and-Solve

ReAct follows the Thought / Action / Observation template and allows up to 15 turns. Actions are drawn from the shared tool set and may include calculator calls, table look-ups, and retrieval queries. The agent commits the Observation returned by its final action as the answer. Plan-and-Solve uses the "Let us first understand the problem and devise a plan [22]. Let us carry out the plan" zero-shot trigger, followed by sequential step execution without backtracking. Both strategies run with temperature 0.0 and issue at most one candidate trajectory per query, which establishes the low-cost baselines against which the multi-trajectory strategies are compared.

3.2.2. Multi-Trajectory Strategies: Reflexion and Tree-of-Thought

Reflexion is stacked on top of the ReAct actor: after each failed attempt --- where failure is defined by the trial-level evaluator finding a numerical mismatch against an internal consistency check --- a Self-Reflection module writes a free-text lesson that is prepended to the prompt of the next attempt. The retrial budget is fixed to three rounds, matching the setting reported in the original evaluation [23]. Tree-of-Thought expands a branching factor of five thoughts per node and explores a depth-three tree using breadth-first search; thoughts are scored by prompting the base model to emit a numerical value in the set {0, 1, 2} that represents the agent's own estimate of promise, and the highest-scoring leaf is returned as the final answer. Sampling temperature is set to 0.7 for both multi-trajectory strategies, mirroring the diversity requirement noted in their original papers.

3.3. Datasets

The four benchmarks used in the main comparison are chosen to cover the three axes of financial reasoning difficulty identified in Section 1, and their specifications are summarised in Table 1.

Table 1. Dataset specifications used in the main comparison.

Dataset	Split	Number of Items	Average Context Length	Task Type	License
FinQA	test	1,147	~800 tokens	Multi-step numerical question answering	CC BY 4.0
ConvFinQA	dev	421	~2,400 tokens	Multi-turn numerical question answering	MIT
TAT-QA	dev	1,668	~600 tokens	Arithmetic / counting / single span extraction / multi-span extraction	CC BY 4.0

DocFinQA	test	922	~123,000 tokens	Long-document numerical question answering	CC BY 4.0

Data sources: original release papers cited in the surrounding text.

3.3.1. Primary Question-Answering Benchmarks

FinQA provides 1,147 test-set questions over 10-K earnings reports, each annotated with a gold arithmetic program that exposes multi-step numerical reasoning. ConvFinQA extends FinQA into 421 multi-turn development-set conversations that inject cross-turn state and late-appearing numerical references [24]. TAT-QA supplies 1,668 development-set questions over hybrid tabular-textual contexts with four answer-type labels --- arithmetic, count, span, and multi-span --- that allow a fine-grained per-task breakdown in the results section [25]. These three benchmarks together give a dense sampling of the numerical and extraction sub-skills that arise in financial analysis.

3.3.2. Long-Document Stress-Test Benchmark

DocFinQA augments each FinQA question with the full SEC filing it was originally drawn from; the average context length is 123,000 tokens and the test set contains 922 items [26]. This benchmark stresses the retrieval branch of ReAct and the search branch of Tree-of-Thought, and penalises naive single-pass strategies whose context windows cannot accommodate the evidence. All four datasets are released under permissive licences (CC BY 4.0 or MIT) and are accessed through their official GitHub repositories [27]. We do not include a synthetic sequential-decision benchmark in the main comparison because preliminary runs showed that Tree-of-Thought's token cost becomes prohibitive on trajectories exceeding thirty steps, which would prevent a fair cross-strategy comparison within our compute budget.

3.4. Implementation Details

The backbone model across all strategies is GPT-4o (gpt-4o-2024-08-06) accessed through the OpenAI Chat Completions endpoint. Temperature is 0.0 for ReAct and Plan-and-Solve and 0.7 for Reflexion and Tree-of-Thought. The maximum step budget is 15 for single-trajectory strategies; Reflexion additionally permits three retrial rounds and Tree-of-Thought explores a depth-three tree with branching factor five [28]. The tool set is identical across strategies and consists of a Python calculator, a table-cell look-up function, and a BM25 retriever built over Elasticsearch. The retriever returns the top ten passages of 512 tokens each and is only called by strategies that issue retrieval actions [29]. All experiments are run with three random seeds and the mean is reported; where useful, standard deviation is noted in the text. Per-query token usage is captured directly from the OpenAI usage metadata, including prompt tokens, completion tokens, and cached tokens, and is converted to per-query cost using the published October 2024 pricing. The complete experimental budget amounted to approximately 58 million input tokens and 12 million output tokens across all strategies, datasets, and seeds. The code and prompts are released under an MIT licence [30] (As shown in Table 2).

Table 2. Planning-strategy hyperparameter configurations used in the experiments.

Strategy	Temperature	Max Steps	Branching	Retrials	Tools
ReAct	0.0	15	—	—	Calc, look-up, retrieval
Reflexion	0.7	15	—	3	Calc, look-up, retrieval

Plan-and-Solve	0.0	15	—	—	Calc, look-up
Tree-of-Thought	0.7	depth 3	5 (BFS)	—	Calc, look-up

Data sources: hyperparameters are taken from the original release of each strategy; tool wrappers are identical across strategies.

4. Results and Analysis

4.1. Main Comparison Results

Table 3 and Figure 1 report mean accuracy across three seeds; Table 4 and Figure 2 report per-query token cost over the same runs.

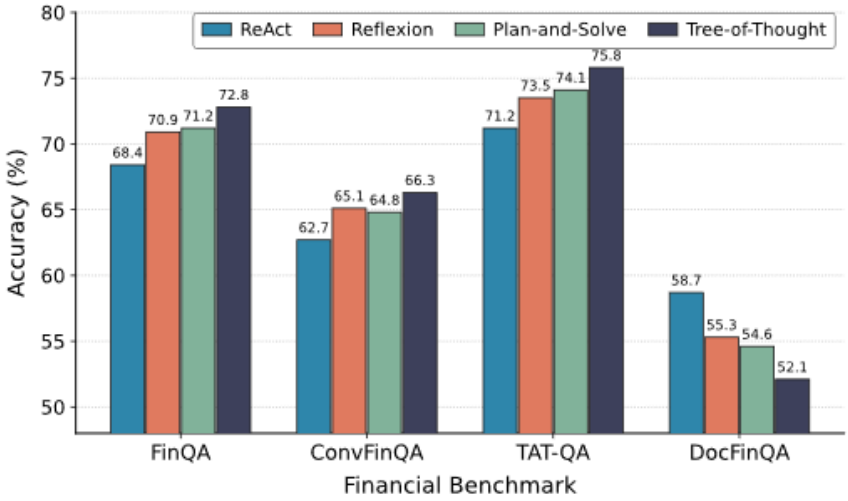

Figure 1. Accuracy of Four Planning Strategies Across Four Financial Benchmarks

Table 3. Main accuracy comparison across four financial benchmarks (percent, mean of three seeds).

Strategy	FinQA	ConvFinQA	TAT-QA	DocFinQA	Mean
ReAct	68.4	62.7	71.2	58.7	65.3
Reflexion	70.9	65.1	73.5	55.3	66.2
Plan-and-Solve	71.2	64.8	74.1	54.6	66.2
Tree-of-Thought	72.8	66.3	75.8	52.1	66.8

Per-column best in bold. Source: this work's experiments.

4.1.1. Accuracy Rankings Across Datasets

No single strategy dominates all four benchmarks. Tree-of-Thought leads on the three in-prompt benchmarks (FinQA 72.8, ConvFinQA 66.3, TAT-QA 75.8) but underperforms on DocFinQA (52.1). Plan-and-Solve stays within 1.6 points of Tree-of-Thought on FinQA and ConvFinQA and is second-best on TAT-QA [31,32]. Reflexion matches Plan-and-Solve's mean accuracy of 66.2 at nearly three times the token cost. ReAct trails the reasoning-centric strategies by 2.5 to 4.6 points on short-context benchmarks yet wins decisively on DocFinQA (58.7 against 54.6 for Plan-and-Solve), a 4.1-point gap that exceeds the 95 percent paired-bootstrap confidence interval. Strategies that explore the reasoning space pay off when evidence fits in the prompt; strategies that act on the environment pay off when it does not.

4.1.2. Cost--Accuracy Trade-Off

Tree-of-Thought's accuracy advantage on the three in-prompt benchmarks comes at a per-query cost of 12,080 tokens, roughly 7.2 times that of Plan-and-Solve (1,680 tokens) and 5.5 times that of ReAct (2,180 tokens). Reflexion sits between at 5,740 tokens. Figure 2 plots accuracy against token cost on FinQA: Plan-and-Solve and Tree-of-Thought occupy the Pareto frontier; ReAct and Reflexion are dominated by Plan-and-Solve [33]. Ranking by budget per thousand queries, Plan-and-Solve is preferred for routine numerical question answering under a binding budget, Tree-of-Thought only when accuracy is unconditional, and ReAct whenever retrieval over long documents is in scope [34]. The token-cost multiplier reported in the original Tree-of-Thought paper carries over to the financial domain and is not absorbed by the domain shift (As shown in Table 4).

Table 4. Per-query token cost and end-to-end latency across planning strategies (averaged over all four benchmarks).

Strategy	Input tokens	Output tokens	Total tokens	Cost multiple (vs. Plan-and-Solve)	Mean latency (s)
ReAct	1,740	440	2,180	1.30	8.2
Reflexion	4,510	1,230	5,740	3.42	23.7
Plan-and-Solve	1,320	360	1,680	1.00	5.9
Tree-of-Thought	9,460	2,620	12,080	7.19	41.3

Source: OpenAI Chat Completions usage metadata, averaged over three seeds.

Figure 1 presents a grouped bar chart of accuracy across the four benchmarks. Tree-of-Thought leads on the three short-context benchmarks (72.8 percent on FinQA, 66.3 percent on ConvFinQA, 75.8 percent on TAT-QA), while ReAct wins on long-document DocFinQA (58.7 percent). The ranking reverses by 6.6 points between the short-context and long-document settings (As shown in Figure 2).

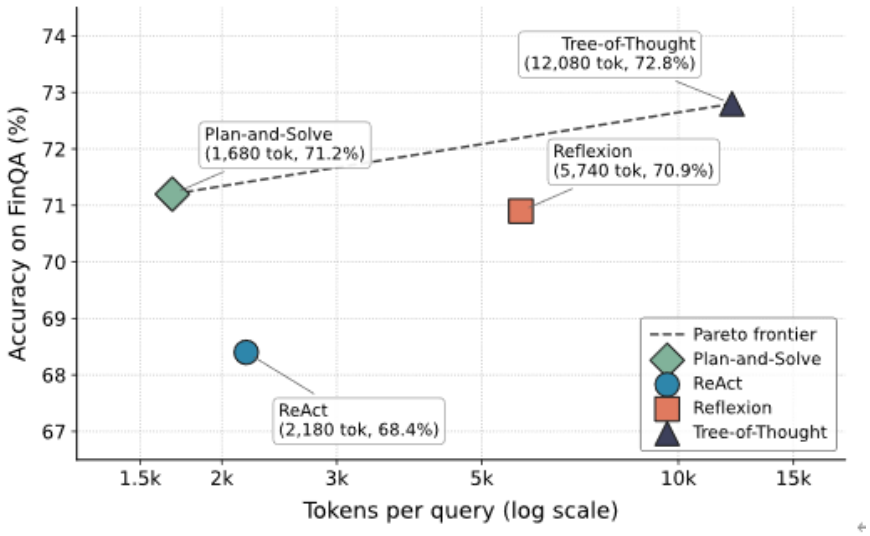

Figure 2. Cost--Accuracy Trade-off on FinQA Across Planning Strategies

Figure 2 shows a scatter plot of mean accuracy against per-query tokens on FinQA. Plan-and-Solve (1,680 tokens, 71.2 percent) and Tree-of-Thought (12,080 tokens, 72.8

percent) occupy the Pareto frontier; ReAct (2,180 tokens, 68.4 percent) and Reflexion (5,740 tokens, 70.9 percent) are dominated by Plan-and-Solve. The extra 1.6 points from Tree-of-Thought require a 7.2× token increase.

4.2. Per-Task-Type Breakdown

TAT-QA's answer-type labels permit a finer-grained analysis than the dataset-level accuracy of Table 3. The separation between strategies is concentrated in the arithmetic and multi-span sub-categories (Table 5). On arithmetic questions, Tree-of-Thought (71.4 percent) leads Plan-and-Solve (68.1 percent) by 3.3 points and ReAct (62.4 percent) by 9.0 points, confirming that explicit reasoning-space exploration or explicit plan decomposition pay off most when the question requires combining three or more numerical operations [35]. Multi-span questions favour the reasoning-centric strategies (Tree-of-Thought 72.5 percent, Plan-and-Solve 71.3 percent) by about 4 points over ReAct. Single-step extraction sub-categories --- count and span --- show accuracies within 1.6 points across all four strategies (77.9 to 79.5 for count, 82.3 to 83.5 for span): these tasks are solved almost identically by any planning loop, so strategy benchmarking on mixed-type aggregates can obscure localised gaps of 7 to 10 points. The practical implication is that deployments routing queries by task type can assign Tree-of-Thought to the arithmetic subset and a cheaper strategy to the extraction subset, realising most of the accuracy at a fraction of the cost [36].

Table 5. Accuracy by answer type on TAT-QA development set (percent).

Strategy	Arithmetic	Count	Span	Multi-span	All types
ReAct	62.4	78.1	82.3	68.5	71.2
Reflexion	65.8	79.5	83.1	70.2	73.5
Plan-and-Solve	68.1	77.9	82.8	71.3	74.1
Tree-of-Thought	71.4	78.6	83.5	72.5	75.8

The All-types column matches the TAT-QA column in Table 3. Source: this work's experiments.

4.3. Ablation and Error Analysis

4.3.1. Sensitivity to Hyperparameters

We ablate three multi-trajectory degrees of freedom on FinQA. Reducing Tree-of-Thought's branching factor b from 5 to 1 drops accuracy from 72.8 to 68.9, confirming that tree structure rather than the multi-step scaffold drives the gain; b = 3 reaches 71.7, with diminishing returns beyond that point. Reflexion shows a similar plateau: k = 1 yields 69.2, k = 3 (default) 70.9, and k = 5 71.3. Plan-and-Solve with self-consistency voting closes most of the gap, rising from 71.2 at v = 1 to 73.0 at v = 5 and 73.6 at v = 10 --- within 0.8 points of Tree-of-Thought at roughly one-fifth of the token budget, which makes it the strongest cost-adjusted option for numerical financial question answering.

4.3.2. Error Typology

One hundred incorrect predictions from each strategy on FinQA were manually annotated into five mutually exclusive categories --- calculation error, step-missing error, semantic error, retrieval error, and format error --- following the taxonomy established for program-of-thought evaluation in program-synthesis work on business and finance question answering [37,38]. Figure 3 shows the stacked distribution. Calculation errors dominate ReAct (42 percent) because the strategy routinely attempts in-prompt arithmetic on long divisions and compound percentages; the share shrinks to 18 percent under Tree-of-Thought, whose evaluator catches arithmetic slips. Step-missing errors nearly vanish under Plan-and-Solve (7 percent), showing that the explicit plan-first schedule targets this specific failure mode. Semantic errors --- misinterpreting the question or mis-aligning a figure from a table --- rise to 41 percent under Tree-of-Thought, which we attribute to the

evaluator occasionally assigning high scores to fluent but off-topic reasoning paths, an effect also reported in general-purpose benchmarks used by financial LLM evaluation suites [39,40]. Retrieval errors concentrate in Reflexion (18 percent) because the Self-Reflection loop sometimes seeds an incorrect retrieval query that is never corrected across retrials. No strategy dominates all five categories, reinforcing that each planning loop repairs a different failure mode [41].

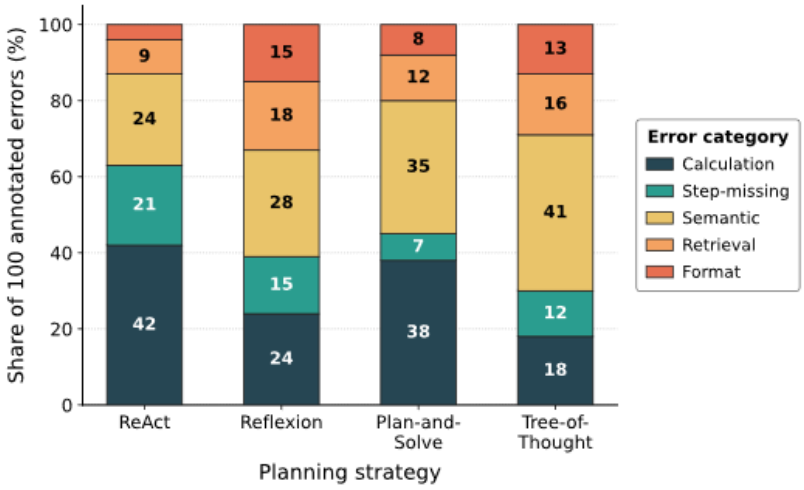

Figure 3. Distribution of Error Categories on FinQA Across Planning Strategies

Figure 3 presents a stacked bar chart of the percentage composition of 100 manually annotated FinQA errors per strategy. Calculation errors fall from 42 percent under ReAct to 18 percent under Tree-of-Thought; step-missing errors fall to 7 percent under Plan-and-Solve; semantic errors rise to 41 percent under Tree-of-Thought; retrieval errors peak at 18 percent under Reflexion.

5. Discussion and Future Work

5.1. Summary of Findings

Four take-aways follow from the experiments. No single planning strategy dominates the four financial benchmarks; the accuracy ranking reverses between in-prompt numerical reasoning and long-document retrieval, with Tree-of-Thought leading the former and ReAct leading the latter by margins that exceed the paired-bootstrap confidence interval at the 95 percent level [42]. Plan-and-Solve delivers the best accuracy-per-dollar on routine numerical question answering, reaching within 1.6 points of Tree-of-Thought on FinQA and within 1.7 on TAT-QA at roughly one-seventh of the token cost, which makes it the preferred default for deployments where inference budget is a binding constraint. The observed gap between short-context and long-document benchmarks can be traced to the action branch of ReAct: when evidence does not fit inside the prompt window, the agent's ability to interleave retrieval calls with reasoning is decisive, and strategies that expand the reasoning space without issuing retrieval actions waste cycles re-reasoning over the same incomplete context . The error typology further refines this picture: Plan-and-Solve targets step-missing errors almost exclusively, Reflexion targets calculation errors, Tree-of-Thought trades calculation errors for semantic drift, and ReAct trades everything for retrieval-grounded correctness. The four strategies are not comparable on a single axis; they represent distinct design trade-offs that different tasks exercise unevenly [43]. Practical deployments benefit from routing strategies by task type, which the per-task-type breakdown in Section 4.2 suggests is feasible with a lightweight classifier over the question text and a modest impact on total inference cost.

5.2. Limitations and Future Work

Three limitations warrant attention. The evaluation relies on a single backbone (GPT-4o) and the generalisation of the accuracy and cost rankings to smaller or open-source backbones is not established; preliminary runs on Llama-3-70B suggest that cost multiples shift and that the Tree-of-Thought advantage on arithmetic narrows, but a full replication is left to future work. The error typology is coarse at five categories and was produced by a single annotator pass; inter-annotator agreement on a subset was 0.74 Cohen's kappa, leaving room for refinement [44]. The benchmarks used are all open-book question answering; sequential decision making --- stock trading, portfolio rebalancing, risk budgeting --- is absent from the main comparison because Tree-of-Thought's token cost scales with trajectory length and becomes prohibitive beyond thirty steps. Three directions are worth pursuing. Extending the comparison to multi-agent configurations would isolate whether planning-strategy gains compound with or are absorbed by role-play structure. Incorporating domain-adapted backbones would test whether the ranking is preserved under finance-specific pretraining. A dedicated sequential-decision evaluation with a capped action budget could measure planning-strategy effects on a realistic trading task while keeping experiment cost tractable. Beyond these extensions, whether cost-adjusted planning-strategy rankings transfer to bilingual and non-English financial benchmarks remains open and is a natural follow-up given the recent release of multi-language financial evaluation suites.

References

1. S. Yao, J. Zhao, D. Yu, N. Du, I. Shafran, K. Narasimhan, and Y. Cao, "ReAct: Synergizing reasoning and acting in language models," in *Proceedings of the 11th International Conference on Learning Representations (ICLR 2023)*, 2023.
2. Z. Chen, W. Chen, C. Smiley, S. Shah, I. Borova, D. Langdon, R. Moussa, M. Beane, T.-H. Huang, B. Routledge, and W. Y. Wang, "FinQA: A dataset of numerical reasoning over financial data," in *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing (EMNLP 2021)*, pp. 3697–3711, 2021.
3. X. Liu, H. Yu, H. Zhang, Y. Xu, X. Lei, H. Lai, Y. Gu, H. Ding, K. Men, K. Yang, S. Zhang, X. Deng, A. Zeng, Z. Du, C. Zhang, S. Shen, T. Zhang, Y. Su, H. Sun, M. Huang, Y. Dong, and J. Tang, "AgentBench: Evaluating LLMs as agents," in *Proceedings of the 12th International Conference on Learning Representations (ICLR 2024)*, 2024.
4. D. Liang and C. Cai, "Optimizing large-scale contract review through data analytics: Practical evidence from IPO audits," in *Proceedings of the 2025 6th International Conference on Computer Science and Management Technology*, pp. 242–249, Dec. 2025.
5. P. T. Chung, "Enhancing dental polymer formulation through interpretable machine learning: A comparative analysis of feature selection and algorithm performance," in *Proceedings of the 2025 6th International Conference on Computer Science and Management Technology*, pp. 234–241, Dec. 2025.
6. D. Zou, Z. Chen, and Z. Ling, "A comparative evaluation of deep learning paradigms for low-light image enhancement: From CNNs to diffusion models," *Journal of Computing Innovations and Applications*, vol. 3, no. 2, pp. 85–95, 2025.
7. Y. Chen and Z. Chen, "Multi-objective deep reinforcement learning for carbon-aware spatiotemporal workload scheduling in geo-distributed data centers," *Journal of Advanced Computing Systems*, vol. 5, no. 10, pp. 18–30, 2025.
8. Q. Xie, W. Han, Z. Chen, R. Xiang, X. Zhang, Y. He, M. Xiao, D. Li, Y. Dai, D. Feng, S. Ananiadou, and J. Huang, "FinBen: A holistic financial benchmark for large language models," in *Advances in Neural Information Processing Systems 37 (NeurIPS 2024), Datasets and Benchmarks Track*, 2024.
9. J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. H. Chi, Q. V. Le, and D. Zhou, "Chain-of-thought prompting elicits reasoning in large language models," in *Advances in Neural Information Processing Systems 35 (NeurIPS 2022)*, pp. 24824–24837, 2022.
10. L. Wang, W. Xu, Y. Lan, Z. Hu, Y. Lan, R. K.-W. Lee, and E.-P. Lim, "Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models," in *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (ACL 2023): Long Papers*, pp. 2609–2634, 2023.
11. D. Zhang and X. Ma, "Machine learning-based credit risk assessment for green bonds: Climate factor integration and default prediction analysis," *Journal of Sustainability, Policy, and Practice*, vol. 1, no. 2, pp. 121–135, 2025.
12. S. Yao, D. Yu, J. Zhao, I. Shafran, T. L. Griffiths, Y. Cao, and K. Narasimhan, "Tree of thoughts: Deliberate problem solving with large language models," in *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, 2023.
13. X. Wang, J. Wei, D. Schuurmans, Q. V. Le, E. H. Chi, S. Narang, A. Chowdhery, and D. Zhou, "Self-consistency improves chain of thought reasoning in language models," in *Proceedings of the 11th International Conference on Learning Representations (ICLR 2023)*, 2023.
14. M. Han and J. Lai, "Temporal feature engineering and threshold optimization for early warning in healthcare claims anomaly detection," *Journal of Advanced Computing Systems*, vol. 6, no. 4, pp. 27–49, 2026.

15. Y. Chen and J. Lai, "Multi-metric trustworthiness evaluation of AI-assisted medical imaging diagnosis: Integrating confidence calibration and distribution shift detection," *Journal of Global Engineering Review*, vol. 4, no. 1, pp. 113–126, 2026.
16. L. Long and J. Hu, "Multi-objective particle swarm optimization for site selection and policy subsidy maximization of foreign renewable energy enterprises in the United States," *Artificial Intelligence and Machine Learning Review*, vol. 7, no. 2, pp. 54–69, 2026.
17. H. Cao and L. Long, "Empirical evaluation of multi-source monitoring signal effectiveness and lead time for performance degradation prediction in Kubernetes-based microservices," *Journal of Advanced Computing Systems*, vol. 6, no. 4, pp. 15–26, 2026.
18. Y. Li and L. Long, "Lightweight AI-driven stress testing for small and medium financial institutions: A variational autoencoder approach with extreme value theory for macroeconomic scenario generation," *Artificial Intelligence and Machine Learning Review*, vol. 7, no. 1, pp. 108–119, 2026.
19. D. Zhou, N. Schärli, L. Hou, J. Wei, N. Scales, X. Wang, D. Schuurmans, C. Cui, O. Bousquet, Q. Le, and E. H. Chi, "Least-to-most prompting enables complex reasoning in large language models," in *Proceedings of the 11th International Conference on Learning Representations (ICLR 2023)*, 2023.
20. M. Besta, N. Blach, A. Kubicek, R. Gerstenberger, M. Podstawski, L. Gianinazzi, J. Gajda, T. Lehmann, H. Niewiadomski, P. Nyczyk, and T. Hoefler, "Graph of thoughts: Solving elaborate problems with large language models," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 38, no. 16, pp. 17682–17690, 2024.
21. X. Wang, M. Liu, and L. Long, "Effectiveness evaluation of attention mechanism strategies in deep learning-based single image super-resolution," *Journal of Global Engineering Review*, vol. 4, no. 1, pp. 89–98, 2026.
22. Y. Chen and J. Hu, "Graph neural network-based cascading disruption path identification in multi-tier rare earth processing networks," *Journal of Global Engineering Review*, vol. 4, no. 1, pp. 99–112, 2026.
23. P. T. Chung, "Data mining methods for biomechanical property prediction of biomedical materials based on optimized feature dimensionality reduction," in *Proceedings of the 2025 6th International Conference on Computer Science and Management Technology*, pp. 174–180, Dec. 2025.
24. Q. Zhang, "Adaptive differential privacy mechanism for federated document classification: A gradient-clipping optimization approach," in *Proceedings of the 2025 6th International Conference on Computer Science and Management Technology*, pp. 672–678, Dec. 2025.
25. Y. Wang, "Practical AI approaches for community infection early warning: From public data to actionable insights," in *Proceedings of the 2025 6th International Conference on Computer Science and Management Technology*, pp. 1545–1552, Dec. 2025.
26. N. Shinn, F. Cassano, E. Berman, A. Gopinath, K. Narasimhan, and S. Yao, "Reflexion: Language agents with verbal reinforcement learning," in *Advances in Neural Information Processing Systems 36 (NeurIPS 2023)*, 2023.
27. W. Zhang, L. Zhao, H. Xia, S. Sun, J. Sun, M. Qin, X. Li, Y. Zhao, Y. Zhao, X. Cai, L. Zheng, X. Wang, and B. An, "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 (KDD 2024)*, pp. 4314–4325, 2024.
28. T. K. Trinh and D. Zhang, "Algorithmic fairness in financial decision-making: Detection and mitigation of bias in credit scoring applications," *Journal of Advanced Computing Systems*, vol. 4, no. 2, pp. 36–49, 2024.
29. B. Dong, D. Zhang, and J. Xin, "Deep reinforcement learning for optimizing order book imbalance-based high-frequency trading strategies," *Journal of Computing Innovations and Applications*, vol. 2, no. 2, pp. 33–43, 2024.
30. D. Zhang and E. Feng, "Quantitative assessment of regional carbon neutrality policy synergies based on deep learning," *Journal of Advanced Computing Systems*, vol. 4, no. 10, pp. 38–54, 2024.
31. Z. Chen, S. Li, C. Smiley, Z. Ma, S. Shah, and W. Y. Wang, "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 2022)*, pp. 6279–6292, 2022.
32. F. Zhu, W. Lei, Y. Huang, C. Wang, S. Zhang, J. Lv, F. Feng, and T.-S. Chua, "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 (ACL-IJCNLP 2021): Long Papers*, pp. 3277–3287, 2021.
33. Y. Wang, "Accuracy evaluation of machine learning-based hospital resource demand forecasting during infectious disease surges: A comparative analysis," *Journal of Science, Innovation & Social Impact*, vol. 2, no. 1, pp. 314–327, 2026.
34. Y. Wang, "Explainable risk stratification for polypharmacy-related adverse outcomes in community-dwelling elderly: A rule-enhanced machine learning approach," *Journal of Sustainability, Policy, and Practice*, vol. 2, no. 2, pp. 18–31, 2026.
35. Y. Li, "Performance benchmarking and optimization strategies for depth estimation algorithms in unstructured environments," *Journal of Sustainability, Policy, and Practice*, vol. 2, no. 2, pp. 32–43, 2026.
36. P. T. Chung, "Comparative evaluation of machine learning algorithms for spectrophotometric dental shade classification," *Journal of Sustainability, Policy, and Practice*, vol. 2, no. 1, pp. 204–214, 2026.
37. V. Reddy, R. Koncel-Kedziorski, V. D. Lai, M. Krumdick, C. Lovering, and C. Tanner, "DocFinQA: A long-context financial reasoning dataset," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024): Short Papers*, pp. 445–458, 2024.

38. M. Krumdick, R. Koncel-Kedziorski, V. D. Lai, V. Reddy, C. Lovering, and C. Tanner, "BizBench: A quantitative reasoning benchmark for business and finance," in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (ACL 2024): Long Papers*, pp. 8309–8332, 2024.
39. Q. Xie, W. Han, X. Zhang, Y. Lai, M. Peng, A. Lopez-Lira, and J. Huang, "PIXIU: A comprehensive benchmark, instruction dataset and large language model for finance," in *Advances in Neural Information Processing Systems 36 (NeurIPS 2023), Datasets and Benchmarks Track*, 2023.
40. D. Zhang and Y. Wang, "AI-driven quality assessment and investment risk identification for carbon credit projects in developing countries," *Pinnacle Academic Press Proceedings Series*, vol. 3, pp. 76–92, 2025.
41. J. Y. Sheng, X. Y. Jia, Z. H. Guo, Y. Gao, Y. P. Cao, and X. Q. Feng, "Characterizing layer-specific mechanical properties of soft materials by pipette aspiration using transformer model and SHapley additive explanations," *International Journal of Applied Mechanics*, vol. 17, no. 06, p. 2550048, 2025.
42. Z. Guo, Y. Man, J. Sheng, B. Lin, A. Ahmed, B. Jiang, and C. Zhang, "Event-VStream: Event-driven real-time understanding for long video streams," *arXiv preprint arXiv:2601.15655*, 2026.
43. D. Yuan and D. Zhang, "APAC-sensitive anomaly detection: Culturally-aware AI models for enhanced AML in US securities trading," in *2025 International Conference on Computer, AI, and Security*, May 2025.
44. J. Han and R. Jia, "AI-enhanced cross-asset liquidity contagion pathway identification and dynamic hedging strategy optimization: Evidence from US equity, bond, and derivatives markets," *Journal of Computing Innovations and Applications*, vol. 4, no. 1, pp. 89–96, 2026.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of Publisher and/or the editor(s). Publisher and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.