

FinEvo-Bench: A Longitudinal Benchmark for Self-Evolving Agents in Professional Financial Workflows

Bo Deng^{1,2*}, Kang Zhou², Lifan Guo², Chongyang Tao^{1†},
Xuanren Chen¹, Chenggang Xie¹, Renzhao Liang¹, Feng Chen², Chi Zhang²

¹Beihang University

²Qwen DianJin Team, Alibaba Cloud Computing

Abstract

Most agent benchmarks evaluate tasks independently and cannot measure whether experience from one task helps with later tasks. Existing self-evolution benchmarks do not jointly cover professional workflows, open-ended deliverables, and multi-aspect evaluation. We introduce FinEvo-Bench, a longitudinal benchmark with 120 real-case-grounded tasks, 20 business scenes across six financial domains. Institution-provided professional procedures define the required operations and constraints. Eligible institution-provided and publicly documented cases supply the task facts. Each scene contains six related but substantively distinct cases that share a professional procedure and a manually reviewed rubric for task quality and financial compliance. We compare four self-evolving agent scaffolds using the same Qwen3.7-Max backbone and three independently shuffled, globally interleaved task streams. Paired non-evolving controls estimate each scaffold’s self-evolution gain from retained experience, while an independent Claude Code scoring agent backed by Claude Opus 4.6 evaluates all outputs. Letta achieves the highest evolved score (91.65) and fewest compliance issues (0.09 per task); Codex achieves the largest self-evolution gain (+19.37). Across scaffolds, the evolving condition raises scores by 9.33–19.37 points and reduces compliance issues by 0.12–0.44 per task. Paired score gains at within-scene ranks 4–6 exceed those at ranks 1–3 by 6.10–8.70 points. In Claude Code, skill-only evolution produces higher task quality and fewer compliance issues than memory-only and combined memory–skill evolution. Across all four scaffolds, rubric feedback also yields higher scores and fewer compliance issues than reference-answer feedback. FinEvo-Bench measures both professional performance and self-evolution ability: how effectively an agent turns prior experience into later improvement.

1 Introduction

Self-evolving agents retain completed interactions as memories, skills, or other persistent state for use in later work (Shinn et al. 2023; Zhao et al. 2024; Wang et al. 2023; Zhang et al. 2026). Their evaluation must test whether experience from one task becomes reusable knowledge for subsequent tasks, not simply whether the agent completes the task at hand. Most agent benchmarks evaluate each task independently. They measure current execution capability but do not

*This work was conducted during an internship at Alibaba Cloud Computing.

†Corresponding author.

Benchmark	Cross-task Evolution	Domain Workflow	Open-ended Artifact	Multi-aspect Evaluation
LifelongAgent Bench	✓			
SEA-Eval	✓			✓
Evo-Memory	✓			✓
StuLife	✓	✓		✓
Frontier-Eng		✓	✓	✓
FinEvo-Bench (Ours)	✓	✓	✓	✓

Table 1: Representative self-evolution benchmarks.

observe how experience accumulates or transfers. Recent self-evolution benchmarks use sequential task streams, situated environments, or iterative artifact optimization to study this process (Zheng et al. 2025; Jiang et al. 2026; Wei et al. 2025; Cai et al. 2025; Chi et al. 2026).

Table 1 compares representative benchmarks; Section 2 covers the broader literature. Here, a domain workflow groups tasks under a shared professional process, an open-ended artifact admits multiple valid outputs, and multi-aspect evaluation goes beyond terminal success. LifelongAgent-Bench, SEA-Eval, and Evo-Memory test ordered-task reuse; StuLife studies persistent situated adaptation; Frontier-Eng measures iterative artifact improvement. Prior work covers these properties separately but does not combine them in recurring professional workflows. Financial cases share procedures, analytical checks, and compliance requirements while differing in their files, risks, and conclusions. Credit reviews, claim analyses, insurance advice, and investment research all recur across clients or institutions: lessons about evidence checking, analytical sequencing, and professional boundaries can transfer, but every new case still requires case-specific calculations and judgment. Financial workflows therefore test whether an agent can reuse professional procedures without copying case-specific solutions. Recent financial benchmarks evaluate professional artifacts (Jin et al. 2026; Kundurthy et al. 2026), but score cases independently rather than testing whether retained experience improves later related work.

Final artifact quality, domain-specific compliance, and im-

provement from retained experience answer different questions. A scaffold may achieve a high final score because its backbone is already strong, or show a large gain while remaining weaker in absolute terms. We report evolved performance and paired within-scaffold gains separately: the former measures final capability; the latter, self-evolution ability.

FinEvo-Bench addresses this setting. We define *self-evolution* as a cross-task process: an agent extracts signals from completed tasks and feedback, updates its persistent state, and applies that state to later tasks. FinEvo-Bench contains 20 business scenes across six financial domains, with six multi-file tasks per scene and 120 tasks in total. Tasks within a scene are related but substantively distinct and are grounded in eligible institution-provided and publicly documented cases. Before release, direct identifiers and sensitive fields are removed or replaced when necessary. Reference professional procedures validated in practice define the required inputs, main steps, expected deliverables, and professional constraints. Two domain experts construct each scene-level rubric and review it against all six tasks. On the 120 deliverables from one complete main-experiment run, the automated rubric judge achieves high absolute agreement with a financial expert ($\text{ICC}(A, 1) = 0.95$; 95% CI: $[0.93, 0.97]$). The evaluation uses three independently shuffled, globally interleaved streams rather than contiguous scene blocks, requiring each scaffold to retain and retrieve relevant experience amid unrelated intervening tasks. Each evolving run is paired with a state-reset control, allowing us to estimate its self-evolution gain from retained experience.

All four scaffolds outperform their state-reset controls on the same three shuffled streams: scores increase by 9.33–19.37 points and compliance issues decrease by 0.12–0.44 per task. Letta has the highest evolved score (91.65), while Codex has the largest self-evolution gain (+19.37). For every scaffold, mean gains over within-scene ranks 4–6 exceed those over ranks 1–3 by 6.10–8.70 points. Additional diagnostics show that skill-only evolution performs best among Claude Code’s experience carriers, reaching 93.71 points and 0.05 compliance issues per task. Rubric feedback exceeds reference-answer feedback by 3.95–7.93 score points and reduces compliance issues for all scaffolds. Every measured capability dimension improves, with the largest gains in report quality. Cross-scene score differences on a five-scene sample remain small and mixed in direction, while scene isolation triggers 0.01–0.05 more compliance issues per task.

Our contributions are:

- FinEvo-Bench provides 120 real-case-grounded, multi-file tasks with open-ended outputs scored for quality and financial compliance.
- A paired protocol uses interleaved streams and state-reset controls to measure cross-task self-evolution gains.
- Shared-backbone experiments compare four scaffolds, experience carriers, feedback forms, capability dimensions, and cross-scene execution.

2 Related Work

Self-evolution. Self-evolving agents use task outcomes to improve later behavior. They may retain reflections or memories (Shinn et al. 2023; Zhao et al. 2024), compile reusable skills or playbooks (Wang et al. 2023; Zhang et al. 2026), or adapt workflows and model policies (Zhang et al. 2024, 2025; Wang et al. 2025). FinEvo-Bench does not assume a particular mechanism. It compares frameworks with different persistent representations and uses paired non-evolving controls to isolate gains attributable to retained experience.

Longitudinal evaluation of evolving agents. Conventional web, computer, and software-engineering benchmarks evaluate tasks independently and do not preserve task-derived experience across episodes (Zhou et al. 2024; Xie et al. 2024b; Jimenez et al. 2024). Recent benchmarks instead expose task sequences or persistent memory so that adaptation beyond isolated episodes can be assessed. LifelongAgent-Bench constructs skill-dependent interactive tasks (Zheng et al. 2025), while SEA-Eval evaluates correlated and orthogonal streams through success and token trajectories (Jiang et al. 2026). Evo-Memory and EvoMemBench organize their protocols around memory retrieval, update, and reuse; SEAGym separates update evidence from held-out transfer and replay assessment (Wei et al. 2025; Wang et al. 2026; Zheng et al. 2026). Table 1 summarizes the benchmarks closest to our design; EvoMemBench and SEAGym provide complementary memory- and transfer-centered protocols. FinEvo-Bench combines recurring real-case-grounded financial workflows, open-ended professional deliverables, and expert-derived rubric evaluation within a globally interleaved stream.

Financial benchmarks. Financial benchmarks cover numerical and document reasoning (Chen et al. 2021; Zhu et al. 2021; Chen et al. 2022; Islam et al. 2023), broad capability suites (Xie et al. 2023, 2024a), and professional analysis, research, tool-use, or spreadsheet tasks (BenYoash et al. 2025; Bigeard et al. 2025; Choi et al. 2025; Jin et al. 2026; Zhu et al. 2026; Kundurthy et al. 2026). FinRpt’s multidimensional evaluation and BlueFin’s granular criteria accommodate professional artifacts with multiple valid forms, but their instances are still scored independently. FinEvo-Bench interleaves related but distinct cases, isolates retained-experience gains, and separately reports quality and compliance.

3 FinEvo-Bench

3.1 Benchmark Scope and Design

FinEvo-Bench contains 20 business scenes across the six financial domains shown in Figure 1. Tasks cover record intake, calculation, cross-file verification, risk assessment, client advice, and report writing.

Each scene contains six substantively distinct cases that share a professional procedure. The 120 tasks contain 775 input files, averaging 6.46 per task (range: 2–11), and request open-ended reports, assessments, or recommendations. Outputs may differ in form but must be supported by the supplied evidence and satisfy the relevant business and compliance

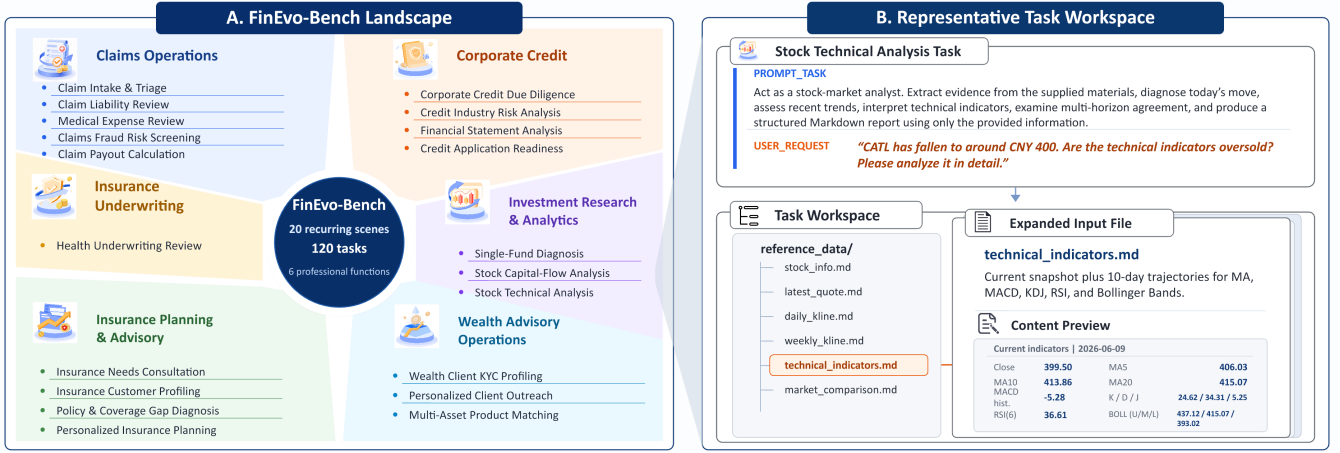


Figure 1: Overview of FinEvo-Bench. Left: 20 business scenes across six financial domains, comprising 120 tasks. Right: an example Stock Technical Analysis task with its natural-language request and six input files.

requirements; scene-level rubrics evaluate these properties without requiring lexical similarity to a reference answer.

3.2 Scene Definition and Task Construction

We construct each scene s from three sources: an institution-provided scene description D_s , a reference professional procedure P_s validated in practice, and a candidate pool C_s drawn from institution-provided and publicly documented cases. The scene description defines the business problem. The reference procedure specifies the required input files, main steps, expected deliverable, and professional constraints. Individual cases supply the facts, data, and judgment conditions. We review source permissions and release conditions before admitting a case to the eligible pool C_s^{eligible} . When necessary, we remove or replace direct identifiers and sensitive fields while preserving data relationships, numerical logic, chronology, and decision conditions.

A domain expert consolidates these sources into a scene specification:

$$S_s = \Phi(D_s, P_s, C_s^{\text{eligible}}) = (O_s, E_s, A_s, Y_s, K_s). \quad (1)$$

Here, O_s is the business objective and scope; E_s , the required inputs; A_s , the professional operations and checks; Y_s , the expected deliverable; and K_s , the compliance requirements and professional boundaries.

After defining a scene, experts select six cases from its eligible pool. The selection follows the scene’s business characteristics and professional procedure. Cases must differ substantively in their business situations, input files, analytical focus, judgment conditions, or conclusions; cases with only superficial differences are excluded from the same scene. Table 2 gives four examples.

Each selected case becomes a task with a natural-language request, input files, task metadata, and a reference answer. Domain experts review the final tasks in two stages. First, they assess the scene and task settings. They then verify that the input files are complete and reflect actual business conditions, and that the reference answer has correct calculations,

Scene	Representative differences across the cases
Financial Statement Analysis	Healthy operations; cyclical downturn; divergence between earnings and cash flow; reporting red flags; multiple distress signals
Claim Payout Calculation	Standard calculation; substantial expense disallowance; ineligible hospitalization; deductible threshold; data anomaly; repeated claims reaching the coverage limit
Single-Fund Diagnosis	High-performing equity fund; persistently underperforming fund; index fund; pure bond fund; mixed bond fund
Personalized Client Outreach	Client-information update; risk reminder; product maturity and rollover; event invitation; routine review; outreach with incomplete client information

Table 2: Examples of six-case design for selected scenes.

analysis, and conclusions. Appendix A gives a construction example with task files, reference answer, rubric, and review.

3.3 Scene-Level Rubrics and Quality Control

The six tasks in a scene share a 100-point rubric derived from the scene’s professional procedure. The rubric checks whether an agent uses the input files correctly, completes the necessary analysis, and reaches supported conclusions. It does not require the output to match the reference answer in structure or wording.

The rubric also checks financial compliance, including fabricated or unsupported data and terms, definitive claims based on insufficient information, decisions beyond the agent’s professional role, and guarantees about credit approval, claim outcomes, or investment returns. Two domain experts construct each rubric. One drafts the criteria, point allocations, and grading rules. The other applies the rubric to all six tasks to check case applicability, coverage of required business steps, and consistency with the input files and refer-

ence answers. They resolve any omission or conflict before using the rubric to score the six tasks. Section 4.2 separately validates the automated application of these rubrics against financial-expert scoring.

4 Experiments

4.1 Experimental Setup

We evaluate four self-evolving agent scaffolds on the 120 tasks and 20 business scenes introduced in Section 3. All scaffolds follow the same longitudinal protocol. Figure 2 summarizes benchmark construction and this protocol.

Self-Evolving Agent Scaffolds. The four scaffolds retain experience in different forms. Claude Code and Codex can distill task experience into reusable skills, project-scoped memory, and global memory. Letta maintains editable memory blocks for each agent. Its memory-management tools can insert, replace, or rewrite content, and every model call receives the full contents of all core memory blocks. GenericAgent distills each completed task into a reusable Markdown experience file. A prompt-resident title-only L0 index guides selective loading of full files (Liang et al. 2026).

Longitudinal Evaluation Protocol. We independently shuffle the 120 tasks three times to form three globally interleaved streams. Each evolving run starts without benchmark-derived experience and processes tasks sequentially in stream order. Each task follows an execute-score-and-feedback-reflect-and-consolidate cycle (Shinn et al. 2023; Zhao et al. 2024; Zhang et al. 2026). After the scaffold produces an answer, a separate Claude Code scoring agent backed by Claude Opus 4.6 applies the scene rubric and returns the rubric feedback defined below. The scaffold reflects on this feedback and may store the resulting experience. We then close the session, removing raw conversation history while preserving stored experience.

The paired non-evolving condition resets agent state before every task. Within each run, both conditions share the task order, backbone, decoding configuration, and scoring procedure; only retention of prior-task feedback differs.

Feedback Visibility and Rubric Isolation. The scaffold never receives the complete scene rubric or the judge’s full item-level record. After finalizing a task, it receives rubric-based feedback summarizing the problems in the current deliverable and their corresponding reasons. The complete evaluation information used by the judge to compute the score is not passed into reflection, while the same scoring procedure applies to both conditions. This design provides concrete feedback for reflection while limiting access to the complete evaluation specification, rather than exposing the full rubric for direct optimization. Because tasks in a scene share a procedure but differ in facts, calculations, and conclusions, later improvement requires generalizing reusable lessons from the current task’s problems and applying them to a new case rather than reproducing an earlier answer. Appendix C gives one complete task example: its deliverable, score, feedback, and retained experience.

Evaluation Metrics. We report four metrics. (1) *Task quality* is the mean scene-specific rubric score on a 0–100 scale. (2) *Financial compliance* is the mean number of triggered compliance issues per task. (3) *Self-evolution ability* is measured by the paired score gain and reduction in compliance issues relative to the non-evolving condition. Let $S_{a,k,i}^{\text{evo}}$ and $S_{a,k,i}^{\text{noevo}}$ denote the scores of scaffold a in run k on task i , and let $C_{a,k,i}^{\text{evo}}$ and $C_{a,k,i}^{\text{noevo}}$ denote the corresponding numbers of compliance issues. For $K = 3$ runs and $N = 120$ tasks,

$$\Delta\text{Score}_a = \frac{1}{KN} \sum_{k=1}^K \sum_{i=1}^N (S_{a,k,i}^{\text{evo}} - S_{a,k,i}^{\text{noevo}}), \quad (2)$$

$$\Delta\text{Comp}_a = \frac{1}{KN} \sum_{k=1}^K \sum_{i=1}^N (C_{a,k,i}^{\text{noevo}} - C_{a,k,i}^{\text{evo}}). \quad (3)$$

(4) *Agent-side cost* is the mean number of tokens consumed during task execution and post-evaluation reflection. We report token counts in units of 10^4 per task. Total tokens combine execution and reflection tokens but exclude the independent rubric judge.

For longitudinal evolution, we rank each scene’s six tasks by their order in the global stream, so rank r follows $r - 1$ same-scene tasks. With $K = 3$ runs, $M = 20$ scenes, and score $S_{a,k,s,r}$, the mean paired gain is

$$\Delta\text{Score}_{a,r} = \frac{1}{KM} \sum_{k=1}^K \sum_{s=1}^M (S_{a,k,s,r}^{\text{evo}} - S_{a,k,s,r}^{\text{noevo}}). \quad (4)$$

We define $\Delta\text{Comp}_{a,r}$ analogously, using the compliance-issue difference $C_{a,k,s,r}^{\text{noevo}} - C_{a,k,s,r}^{\text{evo}}$. Table 3 summarizes ranks 1–3 as Early and ranks 4–6 as Late, with score gains

$$\Delta\text{Score}_a^{\text{early}} = \frac{1}{3} \sum_{r=1}^3 \Delta\text{Score}_{a,r}, \quad (5)$$

$$\Delta\text{Score}_a^{\text{late}} = \frac{1}{3} \sum_{r=4}^6 \Delta\text{Score}_{a,r}. \quad (6)$$

Early and Late compliance reductions use the same rank averages of $\Delta\text{Comp}_{a,r}$. Ranks are induced independently in each run rather than assigned to cases in advance. We mark *experience activation* whenever stored experience enters the execution context. Claude Code, Codex, and GenericAgent retrieve it selectively, whereas Letta injects all core memory blocks automatically. Letta’s 120/120 count therefore denotes an always-on interface, not selective retrieval. We report activation counts and scores for activated and non-activated subsets.

Implementation Details. All four scaffolds use Qwen3.7-Max with a 1M-token context window, a maximum output length of 64K tokens, greedy decoding, and temperature zero. The evaluated Claude Code scaffold also uses Qwen3.7-Max; only the independent Claude Code scoring agent is backed by Claude Opus 4.6. We run each full-benchmark configuration three times and report the mean. The cross-scene diagnostic uses the single sampled 30-task stream described below. Appendix B gives the three-stage pipeline code and prompts.

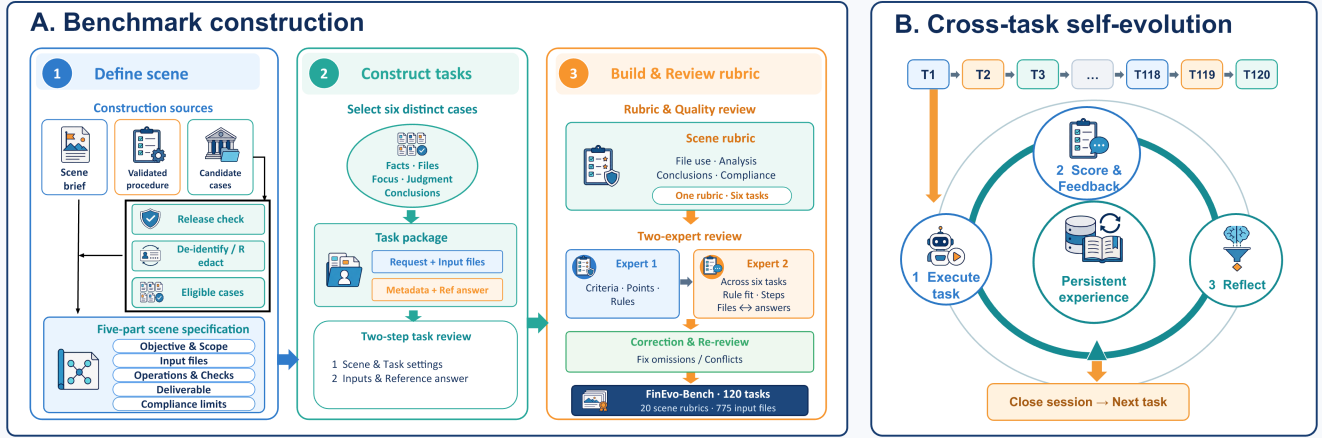


Figure 2: FinEvo-Bench construction and cross-task self-evolution. (A) Dataset construction. (B) Experimental setup.

4.2 Human Validation of Rubric-Based Evaluation

The Claude Code rubric judge and a financial expert independently score the 120 deliverables from one complete main-experiment run using the same scene rubrics. Under a two-way mixed-effects absolute-agreement model, the judge achieves $\text{ICC}(A, 1) = 0.95$ (95% CI: [0.93, 0.97]). ICC here measures agreement in absolute scores, not only consistency in ranking the deliverables. The judge’s mean and maximum absolute differences from expert scores are 1.6 and 5 points, respectively, on the 0–100 scale, supporting its use for the full evaluation.

4.3 Main Results

Overall Performance and Efficiency. Table 3 reports evolved performance and cost, paired gains over controls, and Early/Late gains.

Across the three independently shuffled task streams, retained experience yields positive self-evolution gains for all four scaffolds: relative to the paired non-evolving controls, mean scores rise by 9.33–19.37 points and mean compliance issues fall by 0.12–0.44 per task. Averaged across the three runs, Letta has the highest evolved score (91.65) and fewest compliance issues (0.09 per task), while Codex has the largest self-evolution gain (+19.37). GenericAgent has the lowest evolved score (83.34) and smallest gain (+9.33), despite its lowest token cost. No single scaffold is best in absolute performance, self-evolution gain, and cost.

Letta and GenericAgent use fewer reflection tokens than Claude Code and Codex, but their execution costs differ sharply. Letta uses 17.87×10^4 reflection tokens and 32.56×10^4 execution tokens per task because every core memory block enters each model call. GenericAgent uses only 10.21×10^4 and 11.57×10^4 , respectively, because it writes compact experience files and loads them selectively. Its total cost is less than one-third of Claude Code’s. Agent-side cost depends on both reflection and how stored experience enters the execution context.

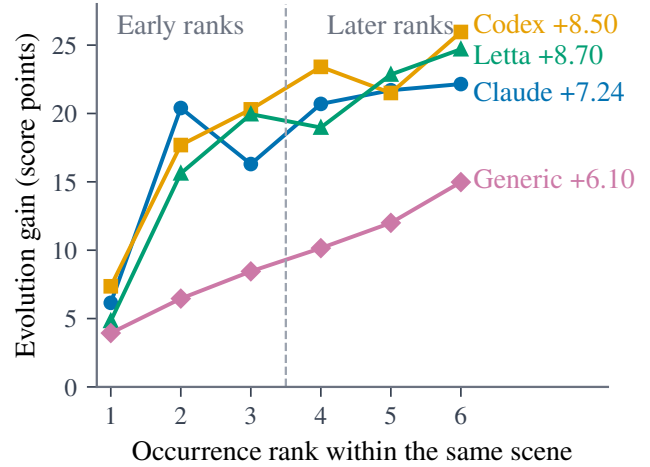


Figure 3: Self-evolution score gain by within-scene occurrence rank, averaged over three runs.

Final scores and self-evolution gains produce different rankings: Letta ranks first by evolved score, whereas Codex ranks first by gain. GenericAgent ranks last on both measures but has the lowest agent-side cost.

Longitudinal Evolution. Despite fluctuations across individual ranks, Figure 3 shows an overall upward trend: every scaffold gains 6.10–8.70 points more over ranks 4–6 than over ranks 1–3. Codex has the highest late gain (23.62), while Letta has the largest early-to-late increase (8.70). Compliance reductions also increase by 0.02–0.10 issues per task (Table 3), supporting gradual experience accumulation. Because later within-scene ranks also occur later globally, the larger gains can reflect both same-scene and broader cross-task experience.

Late gains exceed early gains in each of the three independently shuffled streams, so the trend is not driven by one task order. Each evolving run shares its permutation with the

	Evolved performance		Overall gain		Agent-side cost (10^4 tokens/task)			Longitudinal evolution				
	Agent scaffold	Score $\uparrow$	Comp. $\downarrow$	Δ Score $\uparrow$	Δ Comp. $\uparrow$	Exec. $\downarrow$	Reflect. $\downarrow$	Total $\downarrow$	Score gain		Comp. reduction	
									Early $\uparrow$	Late $\uparrow$	Early $\uparrow$	Late $\uparrow$
Claude Code		89.47	0.11	+17.89	0.44	16.31	60.19	76.50	14.28	21.52	0.43	0.45
Codex		91.17	0.11	+19.37	0.44	20.53	48.22	68.75	15.12	23.62	0.41	0.47
Letta		91.65	0.09	+17.82	0.39	32.56	17.87	50.43	13.47	22.17	0.35	0.43
GenericAgent		83.34	0.34	+9.33	0.12	11.57	10.21	21.78	6.28	12.38	0.07	0.17

Table 3: Performance, cost, and longitudinal evolution across four agent scaffolds.

Agent	Activated	Act. score	Non-act.	Gap
Claude Code	87/120	91.82	83.29	+8.53
Codex	102/120	92.42	84.11	+8.31
Letta	120/120	91.65	–	–
GenericAgent	71/120	87.04	77.98	+9.06

Table 4: Experience utilization during evolution.

Agent scaffold	Δ Score $\uparrow$	Δ Comp. $\downarrow$	Δ Act. $\uparrow$
			(tasks/run)
Claude Code	+6.38	−0.07	+4
Codex	+7.22	− 0.14	+6
GenericAgent	+3.95	−0.10	+9
Letta	+7.93	−0.06	–

Table 6: Rubric feedback versus reference-answer feedback.

Setting	Score $\uparrow$	Comp. $\downarrow$	Cost (10^4 tokens/task)	
			Exec. $\downarrow$	Reflect. $\downarrow$
No evolution	71.58	0.55	14.12	–
Fixed expert skill	86.67	0.13	15.92	–
Full evolution	89.47	0.11	16.31	60.19
Memory only	90.42	0.09	17.94	26.18
Skill only	93.71	0.05	17.53	44.03

Table 5: Memory and skill carrier analysis on Claude Code.

paired control, while different cases occupy the ranks across runs.

Experience Utilization. In Table 4, Gap is the activated-subset score minus the non-activated-subset score. Letta loads all core memory blocks on every task. Codex and Claude Code selectively use skills or project/global memories on 102 and 87 tasks, respectively, while GenericAgent loads a full experience file on 71 tasks. GenericAgent’s lower rate may reflect the limited information in its title-only L0 index for identifying relevant files.

For the three selective-retrieval scaffolds, activated tasks score 8.31–9.06 points higher than non-activated tasks. The non-activated subsets score 77.98–84.11, still 3.97–12.31 points above the corresponding state-reset averages. This suggests that, on some tasks, the scaffolds consider their internal knowledge and task-local files sufficient and therefore do not activate stored experience.

4.4 Discussion

Memory, Skills, and Their Combination. In Claude Code, we compare memory-only, skill-only, and unrestricted memory–skill evolution under the same full-benchmark protocol. No evolution and a fixed expert skill derived from each scene’s reference workflow serve as references. Costs are reported in 10^4 tokens per task; reference settings omit reflection.

Skill-only performs best, scoring 93.71 with 0.05 compliance issues per task (Table 5). On this benchmark, the result favors skills for recurring procedures such as evidence checks, analytical steps, report structures, and compliance constraints. Memory-only remains competitive and improves substantially over no evolution, but it trails skill-only. The combined memory–skill setting improves neither quality nor compliance. Execution traces show that reflection updates both stores, but later tasks often load only one; incomplete loading may partly explain the absence of an additional gain. Execution costs remain similar across the three evolving settings, but reflection costs vary substantially: memory-only is lowest at 26.18, skill-only uses 44.03, and the unrestricted combination uses 60.19. Skill-only obtains the best outcomes with less reflection overhead than the combination. The fixed expert skill outperforms no evolution but trails all dynamically updated carriers, supporting continued updates beyond an initial professional procedure.

Rubric Feedback vs. Reference Answers. Holding all other settings fixed, we compare rubric feedback with a complete reference answer. Each delta is the rubric-feedback condition minus the reference-answer condition: positive Δ Score and Δ Act. favor rubric feedback, while negative Δ Comp. means fewer issues. Δ Act. is the mean logged-activation difference per 120-task run.

Table 6 shows that rubric feedback raises scores by 3.95–7.93 points, reduces compliance issues by 0.06–0.14, and yields 4–9 more activations for selective-retrieval scaffolds. Letta is omitted from the activation comparison because its interface loads memory on every task. A reference answer gives one valid solution but may not identify what the current response is missing or separate case-specific choices from reusable procedures. Rubric feedback instead identifies omitted evidence, incomplete analysis, and compliance failures in the current response.

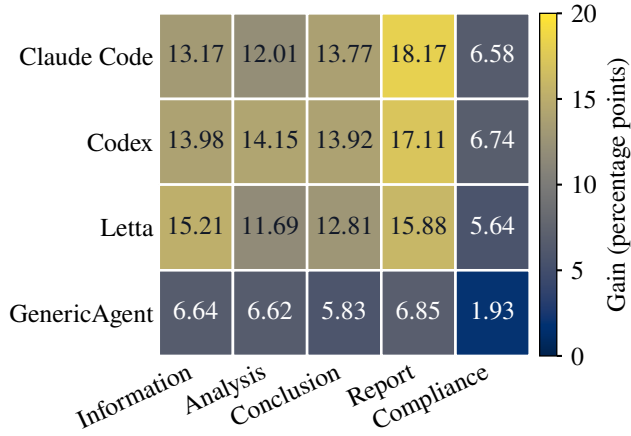


Figure 4: Percentage-point self-evolution gains across five normalized dimensions, averaged over 120 tasks and three runs.

Capability Dimensions. We map every rubric item to one of five dimensions: information and evidence use, analysis and calculation, conclusions and recommendations, report quality, and financial compliance. For a quality dimension d , let $\mathcal{J}_{i,d}$ contain the rubric items assigned to d for task i , with awarded points $p_{a,k,i,j}$ and maximum points $p_{i,j}^{\max}$. Its normalized score rate is

$$R_{a,k,i,d} = \frac{\sum_{j \in \mathcal{J}_{i,d}} p_{a,k,i,j}}{\sum_{j \in \mathcal{J}_{i,d}} p_{i,j}^{\max}}, \quad (7)$$

$$\Delta R_{a,d} = \frac{100}{KN} \sum_{k=1}^K \sum_{i=1}^N (R_{a,k,i,d}^{\text{evo}} - R_{a,k,i,d}^{\text{noevo}}). \quad (8)$$

For compliance, let $c_{a,k,i}$ be the number of triggered issues and $G_{s(i)}$ the total number of compliance issues defined for task i 's scene. We report

$$\Delta R_a^{\text{comp}} = \frac{100}{KN} \sum_{k=1}^K \sum_{i=1}^N \left(\frac{c_{a,k,i}^{\text{noevo}}}{G_{s(i)}} - \frac{c_{a,k,i}^{\text{evo}}}{G_{s(i)}} \right). \quad (9)$$

Quality gains subtract the non-evolving score rate from the evolving score rate. Compliance gain reverses the subtraction, so positive values always indicate fewer normalized issue triggers. All five gains are reported in percentage points.

Every scaffold improves across all five dimensions (Figure 4). Report quality has the largest quality gain for each scaffold (6.85–18.17 percentage points), while normalized compliance gains range from 1.93 to 6.74 points. The comparatively large report-quality gains suggest that structure, coverage, and expression transfer more readily than evidence use, analysis, and conclusions, which remain tied to case-specific facts. GenericAgent improves least in every dimension, consistent with its lower overall gain and less frequent activation.

Cross-Scene Interference. Interleaved scenes may create retrieval competition (Wei et al. 2025; Wang et al. 2026). We randomly sample five scenes (six tasks each) and shuffle their 30 tasks once into a mixed stream. Scene-isolated

Agent scaffold	ΔScore	$\Delta\text{Comp.}$
Claude Code	−0.84	+0.04
Codex	+0.46	+0.01
GenericAgent	+1.63	+0.02
Letta	−1.00	+0.05

Table 7: Scene-isolated execution relative to mixed-scene execution over five sampled scenes (30 tasks).

runs extract each scene’s six tasks in their relative order from that stream and use a fresh agent state for each scene. The conditions use the same tasks and within-scene order; only experience from intervening scenes differs. Let $S_{\text{mixed}}, C_{\text{mixed}}$ and $S_{\text{single}}, C_{\text{single}}$ be the mean score and compliance-issue count over the 30 mixed and isolated outputs, respectively. We report $\Delta\text{Score} = S_{\text{single}} - S_{\text{mixed}}$ and $\Delta\text{Comp.} = C_{\text{single}} - C_{\text{mixed}}$; positive values favor isolation in score but mixed execution in compliance.

Table 7 shows small score differences with no common direction. Scene isolation lowers Claude Code and Letta by 0.84 and 1.00 points, but raises Codex and GenericAgent by 0.46 and 1.63 points. Letta’s always-on memory may benefit from general experience accumulated across scenes, whereas GenericAgent’s title-only index may retrieve more precisely from a homogeneous scene store. Scene-isolated execution also triggers 0.01–0.05 more compliance issues per task for every scaffold, consistent with cross-scene transfer of general practices such as evidence checking, cautious wording, and avoiding decisions beyond the agent’s role. Score effects, however, remain mixed in this five-scene sample.

5 Conclusion and Limitations

FinEvo-Bench evaluates four self-evolving agent scaffolds on 120 real-case-grounded financial tasks. The automated rubric judge agrees closely with financial-expert scoring on 120 outputs ($\text{ICC}(A, 1) = 0.95$). Across three shuffled, globally interleaved streams, all four scaffolds outperform their paired state-reset controls: scores rise by 9.33–19.37 points and compliance issues fall by 0.12–0.44 per task. Letta records the highest evolved score (91.65), whereas Codex records the largest self-evolution gain (+19.37); GenericAgent has the lowest cost. Skill-only evolution performs best among Claude Code’s carriers, rubric feedback outperforms reference answers, and all five capability dimensions improve. Cross-scene score differences remain small and mixed, while scene isolation triggers 0.01–0.05 more compliance issues per task.

FinEvo-Bench covers 20 scenes in six financial domains, uses one backbone, and studies only non-parametric evolution. Results may therefore differ across professions, backbones, and parameter-updating agents. The carrier comparison covers only Claude Code; the cross-scene diagnostic uses five scenes and one ordering. Future work should expand these settings.

References

- BenYoash, N.; Brief, M.; Ovadia, O.; Shenderovitz, G.; Mishaeli, M.; Lemberg, R.; and Sheetrit, E. 2025. SECQUE: A Benchmark for Evaluating Real-World Financial Analysis Capabilities. In *Proceedings of the Fourth Workshop on Generation, Evaluation and Metrics*.
- Bigeard, A.; Nashold, L.; Krishnan, R.; and Wu, S. 2025. Finance Agent Benchmark: Benchmarking LLMs on Real-world Financial Research Tasks. *arXiv preprint arXiv:2508.00828*.
- Cai, Y.; Hao, Y.; Zhou, J.; et al. 2025. Building Self-Evolving Agents via Experience-Driven Lifelong Learning: A Framework and Benchmark. *arXiv preprint arXiv:2508.19005*.
- Chen, Z.; Chen, W.; Smiley, C.; et al. 2021. FinQA: A Dataset of Numerical Reasoning over Financial Data. In *Proceedings of EMNLP*.
- Chen, Z.; Li, S.; Smiley, C.; Ma, Z.; Shah, S.; and Wang, W. Y. 2022. ConvFinQA: Exploring the Chain of Numerical Reasoning in Conversational Finance Question Answering. In *Proceedings of EMNLP*.
- Chi, Y.; Hong, D.; Jiang, D.; et al. 2026. Frontier-Eng: Benchmarking Self-Evolving Agents on Real-World Engineering Tasks with Generative Optimization. *arXiv preprint arXiv:2604.12290*.
- Choi, C.; Kwon, J.; Lopez-Lira, A.; et al. 2025. FinAgent-Bench: A Benchmark Dataset for Agentic Retrieval in Financial Question Answering. *arXiv preprint arXiv:2508.14052*.
- Islam, P.; Kannappan, A.; Kiela, D.; Qian, R.; Scherrer, N.; and Vidgen, B. 2023. FinanceBench: A New Benchmark for Financial Question Answering. *arXiv preprint arXiv:2311.11944*.
- Jiang, S.; Ma, L.; Hong, Z.; et al. 2026. SEA-Eval: A Benchmark for Evaluating Self-Evolving Agents Beyond Episodic Assessment. *arXiv preprint arXiv:2604.08988*.
- Jimenez, C. E.; Yang, J.; Wettig, A.; Yao, S.; Pei, K.; Press, O.; and Narasimhan, K. 2024. SWE-bench: Can Language Models Resolve Real-World GitHub Issues? In *International Conference on Learning Representations*.
- Jin, S.; Li, S.; Zhang, S.; and Yan, R. 2026. FinRpt: Dataset, Evaluation System and LLM-based Multi-agent Framework for Equity Research Report Generation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, 507–515.
- Kundurthy, S.; Na, C.; Moraine, C.; et al. 2026. BlueFin: Benchmarking LLM Agents on Financial Spreadsheets. *arXiv preprint arXiv:2605.30907*.
- Liang, J.; Han, J.; Li, W.; Wang, X.; Zhang, Z.; Jiang, Z.; Liao, Y.; Li, T.; Huang, Y.; Shen, H.; Wu, H.; Guo, F.; Wang, K.; Hong, Z.; Lu, Z.; Ma, L.; Jiang, S.; and Xiao, Y. 2026. GenericAgent: A Token-Efficient Self-Evolving LLM Agent via Contextual Information Density Maximization. *arXiv preprint arXiv:2604.17091*.
- Shinn, N.; Cassano, F.; Berman, E.; Gopinath, A.; Narasimhan, K.; and Yao, S. 2023. Reflexion: Language Agents with Verbal Reinforcement Learning. In *Advances in Neural Information Processing Systems*.
- Wang, G.; Xie, Y.; Jiang, Y.; Mandlekar, A.; Xiao, C.; Zhu, Y.; Fan, L.; and Anandkumar, A. 2023. Voyager: An Open-Ended Embodied Agent with Large Language Models. *arXiv preprint arXiv:2305.16291*.
- Wang, Y.; Zhang, Z.; Chi, M.; et al. 2026. EvoMemBench: Benchmarking Agent Memory from a Self-Evolving Perspective. *arXiv preprint arXiv:2605.18421*.
- Wang, Z.; Wang, K.; Wang, Q.; et al. 2025. RAGEN: Understanding Self-Evolution in LLM Agents via Multi-Turn Reinforcement Learning. *arXiv preprint arXiv:2504.20073*.
- Wei, T.; Sachdeva, N.; Coleman, B.; et al. 2025. EvoMemory: Benchmarking LLM Agent Test-time Learning with Self-Evolving Memory. *arXiv preprint arXiv:2511.20857*.
- Xie, Q.; Han, W.; Chen, Z.; et al. 2024a. FinBen: A Holistic Financial Benchmark for Large Language Models. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*.
- Xie, Q.; Han, W.; Zhang, X.; Lai, Y.; Peng, M.; Lopez-Lira, A.; and Huang, J. 2023. PIXIU: A Comprehensive Benchmark, Instruction Dataset and Large Language Model for Finance. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*.
- Xie, T.; Zhang, D.; Chen, J.; et al. 2024b. OSWorld: Benchmarking Multimodal Agents for Open-Ended Tasks in Real Computer Environments. In *Advances in Neural Information Processing Systems*.
- Zhang, J.; Xiang, J.; Yu, Z.; et al. 2025. AFlow: Automating Agentic Workflow Generation. In *International Conference on Learning Representations*.
- Zhang, Q.; Hu, C.; Upasani, S.; et al. 2026. Agentic Context Engineering: Evolving Contexts for Self-Improving Language Models. In *International Conference on Learning Representations*.
- Zhang, S.; Zhang, J.; Liu, J.; Song, L.; Wang, C.; Krishna, R.; and Wu, Q. 2024. Offline Training of Language Model Agents with Functions as Learnable Weights. In *International Conference on Machine Learning*.
- Zhao, A.; Huang, D.; Xu, Q.; Lin, M.; Liu, Y.-J.; and Huang, G. 2024. ExpeL: LLM Agents Are Experiential Learners. In *AAAI Conference on Artificial Intelligence*.
- Zheng, C.; Xue, C.; Liang, B.; Yang, J.; and Zhang, C. 2026. SEAGym: An Evaluation Environment for Self-Evolving LLM Agents. *arXiv preprint arXiv:2606.17546*.
- Zheng, J.; Cai, X.; Li, Q.; et al. 2025. LifelongAgentBench: Evaluating LLM Agents as Lifelong Learners. *arXiv preprint arXiv:2505.11942*.
- Zhou, S.; Xu, F. F.; Zhu, H.; et al. 2024. WebArena: A Realistic Web Environment for Building Autonomous Agents. In *International Conference on Learning Representations*.
- Zhu, F.; Lei, W.; Huang, Y.; et al. 2021. TAT-QA: A Question Answering Benchmark on a Hybrid of Tabular and Textual Content in Finance. In *Proceedings of ACL-IJCNLP*.
- Zhu, J.; Tian, Y.; Li, B.; Wu, K.; Liang, Z.; Li, J.; Zhang, X.; Guo, L.; Chen, F.; Liu, Y.; and Zhang, C. 2026. FinMCP-Bench: Benchmarking LLM Agents for Real-World Financial

Tool Use under the Model Context Protocol. *arXiv preprint
arXiv:2603.24943*.

A Benchmark Construction: A Worked Financial-Statement-Analysis Scene

This appendix documents the construction process from Section 3 for the *Financial Statement Analysis* scene and Task 3. It traces the scene sources and six-task design to the input files, reference answer, scoring rubric, and review process. During evaluation, the agent receives only the natural-language request and input files.

A.1 Scene Sources, Specification, and Six-Task Construction

Construction sources. Each scene s uses the three sources defined in Section 3: an institution-provided scene description D_s , a reference professional procedure P_s validated in practice, and a candidate pool C_s drawn from institution-provided and publicly documented cases. Table 8 lists their roles in task construction.

Source	Information extracted	Role in construction
Scene description D_s	Business setting, professional roles, objective, and scope	Defines the business problem addressed by the scene
Reference procedure P_s	Required input files, professional operations and checks, expected output, and professional constraints	Specifies how the work is performed and reviewed
Candidate cases C_s	Concrete facts, cross-file relations, risk patterns, judgment conditions, and possible conclusions	Provide the facts and data used to construct the six tasks

Table 8: Construction sources and their roles in the worked scene.

We review the source permissions and release conditions of each candidate case before admitting it to the eligible pool C_s^{eligible} .

Scene specification. A domain expert consolidates the scene description, reference procedure, and eligible cases into the five-part specification defined in Section 3. Table 9 gives the result for Financial Statement Analysis.

Element	Meaning	Instantiation in the worked scene
O_s	Objective and scope	Assess corporate financial health for credit due diligence, post-loan review, large-credit assessment, or risk warning, without replacing the final credit decision
E_s	Required input files	Financial statements, statement notes, audit reports, company information, and industry comparisons provided with the task
A_s	Operations and checks	Verify data quality; calculate and interpret financial indicators; analyze profitability, assets, liabilities, cash flow, and DuPont drivers; scan warning signals; reconcile statements; form a supported conclusion
Y_s	Expected deliverable	A structured financial-health report with calculations, findings supported by source data, a risk rating, and credit or monitoring recommendations
K_s	Constraints	Use only supplied data; mark unavailable information; distinguish facts from inference; avoid unsupported assurances or guarantees

Table 9: Extracted specification for the Financial Statement Analysis scene.

The reference procedure determines the work covered by the tasks and scoring rubric. Closely related steps may be combined, and optional operations apply only when the task requires them. Table 10 maps the procedure to scoring requirements.

Procedure stage	Professional requirement	Task and scoring requirement	Rubric coverage
Input and quality review	Confirm the entity, period, accounting basis, available statements and notes, audit opinion, and material changes in policy or consolidation scope.	Identify the input files provided, mark unavailable information, and report any non-standard audit opinion.	Data quality: 8
Profitability and earnings quality	Calculate profitability ratios and distinguish accounting profit from cash realization and recurring operating performance.	Calculate and interpret the required ratios, explain changes, and compare them with the supplied industry data.	Profitability: 15
Asset and liability quality	Examine receivables, inventory, special assets, debt structure, liquidity, and maturity matching.	Support findings with the relevant statements and notes and explain their business implications.	Asset quality: 12
Cash-flow analysis	Analyze operating, investing, and financing cash flows; calculate free cash flow; interpret their joint pattern.	Cover all three cash-flow categories, operating-cash-flow quality, free cash flow, and the combined pattern.	Cash flow: 13
DuPont decomposition	Decompose ROE and attribute changes to margin, turnover, and leverage.	Assess the decomposition, change attribution, and sustainability of the main driver.	DuPont: 10
Warning review	Scan reporting warning signals and check scene-specific risk conditions.	Check the defined warning signals, possible reporting problems, and the eight risk conditions.	Warning review: 15
Valuation when applicable	Add relative or absolute valuation only when the business request requires it.	Apply valuation criteria only to tasks that require valuation.	Not universally scored
Cross-statement reconciliation	Verify consistency across the balance sheet, income statement, cash-flow statement, and notes.	Check the defined relations and explain any unresolved discrepancy.	Reconciliation: 5
Synthesis and delivery	Produce a health assessment, prioritized risks, quantitative support, monitoring actions, and a professional report.	Assess the conclusion, quantitative support, professional boundaries, and report quality.	Conclusions: 12; report: 10

Table 10: Reference procedure and corresponding scoring requirements. The procedure is not shown to evaluated agents.

Six-task construction. Experts select six cases from the eligible pool based on the scene’s business characteristics and professional procedure. The cases differ in company condition, input files, analytical focus, judgment conditions, or conclusions; cases with only superficial differences are excluded. Table 11 lists the six Financial Statement Analysis cases.

Task	Business setting	Input files	Main analytical focus	Files	Difficulty
1	Healthy mature manufacturer seeking a working-capital facility	Complete three-year statements, audit report, notes, and industry benchmark	Full-process baseline: profitability, solvency, cash flow, DuPont analysis, reconciliation, and credit recommendation	10	Easy
2	Cyclical manufacturer under post-loan review	Complete statements with selected notes and sector benchmark	Separate firm-specific deterioration from an industry downturn; assess efficiency and cash-flow pressure	9	Medium
3	Fast-growing materials company requesting a larger credit line	Complete statements, audit report, revenue and receivable notes, related-party information, non-recurring items, and benchmark	Assess earnings quality, collection risk, leverage-driven ROE, and credit capacity despite headline growth	10	Medium–hard
4	Listed manufacturer under financial-reporting scrutiny	Complete statements plus cash, construction-in-progress, other-receivable, and related-party notes	Detect multiple warning signals, verify cross-statement consistency, and assess the level of risk	10	Hard
5	Trading company with recent delinquency and severe deterioration	Complete statements, non-standard audit report, borrowing, receivable, contingent-liability, and benchmark files	Assess losses, leverage, debt service, negative operating cash flow, and multiple risk conditions	9	Hard
6	Private manufacturer with severely incomplete reporting	One-year income statement and closing balance sheet plus company context	Limit the analysis to the available input files, avoid imputation, identify unavailable analyses, and request follow-up files	3	Easy–medium

Table 11: The six cases used for Financial Statement Analysis. File counts refer to the input files released with each task.

Each selected case becomes a task with a natural-language request, input files, task metadata, and a reviewed reference answer.

All six tasks follow the same professional procedure, but their facts and conclusions come from their own input files.

A.2 Complete Worked Task: Task 3

Task 3 concerns Ruiheng New Materials and its request for a CNY 120 million credit line. The agent receives the following request:

User request. “Ruiheng New Materials is applying for a CNY 120 million credit line. It has been growing quickly: revenue has increased for three consecutive years and profit is also rising. However, its cash flow appears problematic, and operating cash flow was negative last year. Please analyze its financial condition and assess whether the available financial information supports the requested credit amount.”

Table 12 lists the components of Task 3 and their visibility. The agent does not receive the reference procedure, task metadata, reference answer, or scoring rubric.

Component	Contents and purpose	Visible to agent
User request	Business question, requested amount, and initial concern	Yes
Input files	Ten files containing company information, financial statements, notes, an audit report, and comparison data	Yes
Task metadata	Variant type, difficulty, and dataset identifiers	No
Reference answer	One reviewed analysis and conclusion	No
Scoring rubric	Criteria, score bands, and professional and compliance checks used by the judge	No

Table 12: Task 3 components and their visibility to the agent.

The task includes ten input files. Table 13 maps each file group to the required analysis and scoring criteria.

Input-file group	Files	Relevant information and required analysis	Rubric links
Business context	company_overview.md	Provides the requested amount and purpose, banking relationship, growth history, and management explanation.	P9-1–P9-4
Core statements	balance_sheet.md; income_statement.md; cashflow_statement.md	Revenue rises by 25% while operating cash flow falls to CNY –80 million and net receivables rise by 75%; the report must calculate, reconcile, and interpret the divergence.	P2-1–P2-3; P3-1; P4-1–P4-4; P5-1–P5-3; P8-1–P8-2
Audit report	audit_report.md	Provides the audit opinion and the emphasis on receivable collection, which must be considered together with the statements and notes.	P1-2–P1-3; P3-1; P6-1
Statement notes	notes_revenue.md; notes_receivables.md; notes_related_party.md; notes_non_recurring.md	Receivables older than one year reach 34.07%, and non-recurring gains equal 38.18% of net profit; the analysis must consider customer concentration, related-party transactions, allowances, and recurring profit.	P2-2–P2-3; P3-1; P6-1–P6-3
Comparison data	industry_benchmark.md	Industry profitability, earnings-quality, solvency, efficiency, and growth values provide the comparison basis for judging whether headline performance is sustainable.	P2-4; P3-4; P5-3; P9-3

Table 13: Task 3 input-file-to-criterion mapping.

A complete response identifies available and unavailable input files; analyzes profitability, earnings quality, assets, liabilities, cash flow, and DuPont drivers; reviews financial warning signals; reconciles the statements; and gives a quantitatively supported rating and credit recommendation within the stated professional scope.

The reviewed reference answer rates the company as *Concern*. It recommends against the full CNY 120 million request and proposes a CNY 60 million one-year facility with collateral, staged disbursement, and monitoring conditions. The cited reasons are negative operating cash flow, rapidly growing and aging receivables, the material contribution of non-recurring gains to profit, and leverage-driven ROE. Other recommendations receive credit if they are quantitatively supported, address the material risks, and remain within professional scope.

A.3 Scene-Level Rubric and Quality Control

Rubric derivation. The six tasks share a 100-point rubric derived from the professional procedure. It checks whether an agent uses the input files correctly, completes the necessary analysis, and reaches supported conclusions. Outputs need not match the reference answer in structure or wording.

Criterion group	Points
Data confirmation and quality assessment	8
Profitability and earnings quality	15
Asset quality and liability structure	12
Cash-flow analysis	13
DuPont analysis	10
Financial warning and reporting-risk review	15
Cross-statement reconciliation	5
Integrated assessment and conclusions	12
Report quality	10
Total	100

Table 14: Rubric allocation for the Financial Statement Analysis scene.

Table 15 gives representative scoring rules. Applicability depends on the task input files and business request. A required analysis cannot be marked non-applicable when the necessary information is available. Numerical checks use the source data and defined formula, allowing a rounding tolerance below 0.5 percentage points or a clearly stated valid alternative convention.

Criterion	Full credit	Partial or zero credit	Non-applicable rule
P2-2 Earnings quality (5)	The deliverable calculates cash collection, profit-to-cash conversion, and recurring-profit share and interprets them using the defined thresholds.	Two measures: 3; one measure or measures without interpretation: 1; no earnings-quality analysis: 0.	Applicable when the required statement values are present.
P3-1 Receivables (4)	The deliverable analyzes turnover or aging, allowance adequacy, and customer concentration using the available notes.	Material but incomplete analysis: 2; no receivables analysis: 0.	Full credit only when the package provides no receivables detail and the deliverable correctly states that the analysis is unavailable.
P5-2 Change attribution (4)	The deliverable attributes changes in ROE using chain substitution or an equivalent complete method.	Abbreviated attribution: 2; no attribution: 0.	Full credit when only one period is supplied and change attribution is impossible.
P9-4 Decision support (3)	The deliverable gives an actionable credit or monitoring recommendation supported by the task input files and avoids unsupported assurance.	Generic recommendation: 2; no recommendation: 0.	Full credit only when the business request genuinely does not call for a decision recommendation.

Table 15: Representative criterion-level scoring rules.

The rubric also checks for fabricated or unsupported information, conclusions that exceed the available input files, decisions beyond the agent's professional role, and unsupported assurances about credit approval or investment returns.

ID	Check	Type	Observable condition in the final deliverable
RED-1	Fabricated financial data	Data use and compliance	Introduces a financial value, ratio, or audit opinion absent from the task input files.
RED-2	Omitted core analysis	Critical task quality	Omits at least two of profitability analysis, cash-flow analysis, and warning review.
RED-3	Skipped or missed risk checks	Critical professional quality	Omits the R1–R8 review or misses a condition explicitly supported by the task input files.
RED-4	Severe-risk misclassification	Critical professional quality	Gives a favorable rating despite clear high-risk information and provides no corresponding warning.
RED-5	Unsupported imputation	Data use and compliance	Fills missing core financial data by unsupported estimation instead of marking the data unavailable.
RED-6	Missing scope statement	Professional scope	Does not state that the report supports analysis but does not constitute a final credit approval decision.
RED-7	Prohibited assurance	Financial compliance	Guarantees approval or returns, or issues an unsupported directive to invest or lend.

Table 16: Professional and financial-compliance checks in the scene rubric.

Quality control. Domain experts first review the scene and task settings, the completeness of the input files, and the correctness of the reference answers. One expert then drafts the rubric from the scene specification and professional procedure. A second

applies it to all six tasks, checking case applicability, coverage of required business steps, and consistency with the input files and reference answers. The experts correct and re-review any omission or conflict. Table 17 lists these checks.

Stage	Review target	Checks	Action when revision is needed
Source screening	Candidate cases	Source permissions and release conditions	Exclude cases that do not meet the release conditions
Scene definition	$(O_s, E_s, A_s, Y_s, K_s)$	Consistency with the scene description, professional procedure, and cases	Revise the scene specification before selecting tasks
Task construction	Request, input files, metadata, and reference answer	Reasonable task setting, complete task package, correct calculations and conclusions, and substantive variation across cases	Correct the task and repeat the review
Rubric drafting	Criteria and grading rules	Coverage of the required business steps, clear score bands, and appropriate non-applicable conditions	Revise the criterion or grading rule
Rubric review	Rubric applied to all six tasks	Applicability to each case and consistency with the input files and reference answers	Revise the affected task or rubric and review it again

Table 17: Review steps for task construction and scoring.

At evaluation time, the judge receives the agent’s final output and the scoring rubric, but not the execution trajectory or reference answer. The criteria assess only information visible in the final output.

Complete criterion inventory. Tables 18 and 19 list all 32 scored criteria: 27 task criteria and five report-quality criteria. The P7 prefix is omitted because valuation is optional and does not apply to all six tasks.

ID	Criterion	Full-credit condition	Pts.
P1-1	Data completeness	The deliverable identifies the available statements, notes, and audit report and marks missing items without unsupported imputation.	3
P1-2	Audit opinion	The deliverable correctly identifies the audit opinion and prominently reports any non-standard opinion; if no audit report is supplied, it states that the opinion is unverified.	3
P1-3	Policy and scope changes	The deliverable checks material accounting-policy and consolidation-scope changes and explains their effect when the relevant information is available.	2
P2-1	Core profitability metrics	Correctly calculate gross margin, core operating margin, net margin, ROE, and other defined core indicators from the supplied data.	4
P2-2	Earnings quality	Calculate and interpret cash collection, profit-to-cash conversion, and recurring-profit share using the defined thresholds.	5
P2-3	Profitability drivers	Attribute margin changes and analyze material expense-ratio movements rather than merely listing values.	3
P2-4	Industry comparison	Compare the core profitability indicators with the supplied industry benchmark or explicitly state that comparison data are unavailable.	3
P3-1	Receivables	Analyze turnover or aging, allowance adequacy, and customer concentration using the available notes; flag an over-one-year share above 30%.	4
P3-2	Inventory	Analyze inventory turnover, impairment allowance, and consistency with revenue using the available information.	3
P3-3	Goodwill and special assets	Evaluate goodwill or long-running construction in progress and associated impairment or capitalization risk when applicable.	2
P3-4	Debt structure and solvency	Analyze interest-bearing debt, maturity structure, liquidity ratios, and interest coverage.	3
P4-1	Three cash-flow categories	Analyze operating, investing, and financing cash flow and identify the material movements in each.	4
P4-2	Operating-cash-flow quality	Reconcile operating cash flow with reported profit and connect the result to earnings-quality measures.	4
P4-3	Free cash flow	Correctly calculate free cash flow and interpret its trend.	2
P4-4	Joint cash-flow pattern	Identify and explain the combined operating, investing, and financing cash-flow pattern and its credit implications.	3
P5-1	Three-factor DuPont	Correctly decompose ROE into net margin, asset turnover, and equity multiplier.	4
P5-2	Change attribution	Use chain substitution or an equivalent method to attribute changes in ROE.	4
P5-3	Driver sustainability	Identify the dominant ROE driver and assess whether it is sustainable.	2

Table 18: Complete scene-level criterion inventory, Part I: data quality through DuPont analysis (58 points).

ID	Criterion	Full-credit condition	Pts.
P6-1	Warning-signal scan	The deliverable reports checks of at least six defined warning types and assigns a severity supported by the task input files.	5
P6-2	Reporting-risk review	The deliverable checks income, cost, profit, and cash-flow information for signs of aggressive reporting or manipulation.	5
P6-3	Risk conditions R1–R8	The deliverable reports checks of all eight defined risk conditions and surfaces any triggered condition in the report summary.	5
P8-1	Cross-statement reconciliation	Verify the defined relations among cash, profit, equity movements, and statement totals.	3
P8-2	Discrepancy handling	Confirm consistency or identify, localize, and qualify any unresolved discrepancy.	2
P9-1	Financial-health rating	Give a clear rating whose severity is consistent with the preceding analysis.	3
P9-2	Strengths and risks	Summarize concrete, case-specific strengths and risks rather than generic observations.	3
P9-3	Quantitative support	Support qualitative conclusions with values, changes, periods, and relevant comparisons.	3
P9-4	Decision support	Give an actionable credit or monitoring recommendation supported by the task input files and avoid unsupported assurance.	3
FMT-1	Clear presentation	Present the main findings and numerical results clearly; tables may be used when appropriate.	2
FMT-2	Analytical coverage	Cover at least seven of the eight required analytical components. Section titles and ordering may vary.	3
FMT-3	Period and source labels	Mark reporting periods and sources for central data and conclusions.	2
FMT-4	Scope statement	State that the report is for decision support and does not constitute a final credit approval decision.	1
FMT-5	Data fidelity	Use only values supported by the task input files.	2

Table 19: Complete scene-level criterion inventory, Part II: warning review, reconciliation, synthesis, and report quality (42 points).

B Three-Stage Evaluation Pipeline and Prompt Templates

This appendix documents the three-stage harness used in the longitudinal evaluation. Stage 1 executes a task, Stage 2 scores the resulting deliverable, and Stage 3 returns the feedback to the evaluated agent for reflection. Stages 1 and 3 run one of the four evaluated agent frameworks—Claude Code, Codex, Letta, or GenericAgent—whereas Stage 2 always uses the same Claude Code judge. For each task, Stage 3 resumes the exact conversation created in Stage 1.

B.1 Pipeline Overview

Table 20 summarizes the executor, workspace, and output of each stage. The task instruction and user request are concatenated into the Stage 1 message sent to the agent; they are not exposed as files in the agent workspace.

Stage	Function	Executor	Visible workspace	Primary output
1	Task execution	One of four evaluated agent frameworks	input/, containing only the files supplied with the task	model_result.md
2	Rubric scoring	Fixed Claude Code judge	input/, RUBRIC.md, and model_result.md	judge.md
3	Feedback reflection	The same agent framework in the exact Stage 1 conversation	The Stage 1 workspace, with judge.md added beside input/	Updated agent state

Table 20: Roles, workspace visibility, and outputs in the three-stage evaluation pipeline.

Algorithm B.1: Three-stage longitudinal evaluation

Input: ordered task stream $(\tau_1, \dots, \tau_N)$; one of the four evaluated frameworks a ; task rubrics; fixed Claude Code judge J
State: agent state S_a accumulated across the task stream

```

1  for each task  $\tau_i$  in stream order do
2    Create input/; concatenate the task instruction and user request as message  $p_i$ , without adding prompt files to the workspace.
3    Start conversation  $C_i$  with  $a$ , send  $p_i$ , and save the Markdown deliverable as model_result.md.
4    In an isolated workspace containing input/, RUBRIC.md, and model_result.md, invoke  $J$  and save its complete result as the single file judge.md.
5    Add judge.md beside input/ in the Stage 1 workspace.
6    Resume the exact conversation  $C_i$ , send the Stage 3 prompt, and allow  $a$  to update  $S_a$ .
7  end for
```

B.2 Stage 1: Task Execution

Stage 1 Prompt Panel — Execute the Current Task	
Executor	Claude Code, Codex, Letta, or GenericAgent
Conversation	A task conversation that will be resumed unchanged in Stage 3
Message construction	The task instruction and user request are concatenated as text before the message is sent
Visible workspace	input/, containing only the input files required by the task
Required output	model_result.md, a Markdown document generated by the agent
Prompt template	
Complete the task described below using the files in input/. Write the final deliverable as model_result.md in the current directory. After the file has been written, reply only "done."	
[Task instruction]	
[User request]	

Figure 5: Stage 1 prompt and workspace state. The two text components are sent in one message and do not appear as workspace files.

B.3 Stage 2: Fixed Claude Code Scoring

Stage 2 Prompt Panel — Score with the Fixed Claude Code Judge	
Executor	Fixed Claude Code judge
Visible workspace	input/; RUBRIC.md; model_result.md
Scoring target	model_result.md
Scoring standard	RUBRIC.md
Required output	judge.md, the single scoring and feedback file
Prompt template	
You are a strict evaluator. The current workspace contains the original task inputs under input/, the scoring rubric in RUBRIC.md, and the agent deliverable in model_result.md.	
Evaluate model_result.md strictly according to RUBRIC.md. Check every scoring item and support each score and deduction with evidence from the deliverable and, when needed, the task inputs.	
Write the complete scoring result and feedback to judge.md. After the file has been written, reply only "done."	

Figure 6: Stage 2 prompt and workspace state. The fixed Claude Code judge produces one scoring and feedback file.

B.4 Stage 3: Feedback Reflection in the Same Conversation

Stage 3 Prompt Panel — Reflect on Feedback	
Executor	The same one of Claude Code, Codex, Letta, or GenericAgent used in Stage 1
Conversation	The exact Stage 1 task conversation, resumed rather than restarted
Visible workspace	The original workspace, with judge.md added beside input/
New information	judge.md, containing the Stage 2 score and feedback
Result	The agent may update its state and accumulate experience
Prompt template	
The file judge.md contains feedback on your previous response. This is feedback for you; you may update your state and accumulate experience based on it. After you have finished, reply only "done."	

Figure 7: Stage 3 prompt and workspace state. Feedback is returned to the exact conversation used for Stage 1.

C Complete Example of Feedback-Driven Experience Consolidation

This appendix follows one Task 04 trace from the Financial Statement Analysis scene. The task asked the evaluated Codex agent to prepare a risk-oriented analysis of an anonymized biomedical company's 2021–2023 consolidated statements, audit

reports, notes, and industry reference data. The trace contains the Stage 1 deliverable, the fixed judge’s score and feedback, and the changes made to the agent’s two persistent experience carriers during Stage 3. Table 21 maps the archived artifacts to the three-stage protocol in Appendix B.

Stage	Archived artifact		Content used in this example	Result
1	answer.txt		The complete financial-analysis deliverable generated from the task files	369-line Markdown report
2	judge.txt		Section scores, deductions, violation count, and textual feedback	88/100; 0 violations
3	before_MEMORY.txt	and	Persistent feedback-oriented experience before the task and the recorded update	6 change blocks
3	memory_CHANGE.txt			
3	before_skill.txt	and	Reusable financial-analysis procedure before the task and the recorded update	9 change blocks
	SKILL_CHANGE.txt			

Table 21: Artifacts in the complete Task 04 trace. The violation count is reported separately from the report’s internal financial-risk findings.

C.1 Stage 1 Deliverable

The agent produced a complete financial-analysis report rather than a short answer. Its eight sections covered data quality, profitability, asset quality, cash flow, DuPont analysis, financial warnings and possible statement manipulation, three-statement reconciliation, and an overall credit conclusion. Table 22 preserves the full report structure and the principal conclusion of each section.

No.	Report section	Content and principal conclusion
1	Data confirmation and quality	Checked the three statements, audit opinions, key audit matters, and missing supporting materials.
2	Profitability	Calculated core profit and profitability ratios; concluded that margin improvement was offset by rising expense ratios.
3	Asset quality	Examined receivables, inventory, related-party balances, goodwill, construction in progress, and cash holdings.
4	Cash flow	Compared operating, investing, and financing cash flows; identified negative 2023 operating cash flow and increased financing dependence.
5	DuPont analysis	Verified the three-factor identity and attributed the direction of ROE growth mainly to higher leverage.
6	Financial warning and manipulation checks	Examined revenue, cost, asset, and cash-flow signals and identified concentrated risk in assets and cash flow.
7	Three-statement reconciliation	Reconciled statement balances and identified an unexplained 2023 retained-earnings difference.
8	Overall assessment	Assigned a high-risk rating, recommended deferring new credit, and listed eight follow-up documents.

Table 22: Structure of the Stage 1 deliverable.

The deliverable was substantively strong but contained six localized omissions or analytical errors. The excerpts below are translated from the archived output and retain the values used by the judge.

Selected evidence from the Stage 1 deliverable	
Profitability table	“Gross margin: 42.00% (2023), 40.02% (2022), and 40.00% (2021).”
Profit quality	The report explicitly calculated the cash collection ratio and net cash ratio, but did not calculate core profit divided by total profit.
Receivables	The report analyzed turnover days and receivables growth against revenue growth, then noted possible year-end collection arrangements.
Inventory	The report compared turnover days with the industry value and discussed possible understatement or incomplete accounting.
ROE attribution	The report verified the three-factor product and stated that leverage was the main driver, but gave no numerical contribution for each factor.

The report also contained substantial case-specific analysis that received full credit. Table 23 records the main quantitative findings supporting its final disposition.

Dimension	Quantitative evidence in the deliverable	Report conclusion
Profit and cash flow	2023 core-profit margin was 10.79%; operating cash flow was −3.60; free cash flow was −10.10; financing cash inflow was 26.60.	Accounting profit remained positive, but internal cash generation weakened and financing dependence increased.
Asset and leverage risk	Cash/total assets was 48.57%, interest-bearing debt/total assets was 44.00%, other receivables/net assets was 22.1%, and construction in progress/total assets was 10.29%.	High cash coexisted with high debt, related-party balances, and long-running construction projects.
Statement reconciliation	2023 net profit was 12.00, whereas retained earnings increased by only 6.00 and no dividend distribution was reported.	The unexplained 6.00 difference required additional supporting material.
Final disposition	The report assigned an overall D (high-risk) rating and listed eight follow-up documents.	Defer new credit and strengthen monitoring of existing exposure until the major uncertainties are resolved.

Table 23: Principal quantitative findings and final disposition in the Stage 1 deliverable. Monetary amounts are in CNY 100 million.

C.2 Stage 2 Score and Feedback

The fixed judge assigned 88 out of 100 points and recorded no violations. Table 24 gives the complete section-level score. All lost points came from six two-point deductions in profitability, asset quality, and DuPont analysis; the other five sections received full credit.

Rubric section	Score	Maximum	Deduction
Data confirmation and quality assessment	8	8	0
Profitability analysis	9	15	6
Asset-quality analysis	8	12	4
Cash-flow analysis	13	13	0
DuPont analysis	8	10	2
Financial warning and manipulation identification	15	15	0
Three-statement reconciliation	5	5	0
Overall assessment and conclusion	12	12	0
Report quality	10	10	0
Total	88	100	12

Table 24: Complete Stage 2 score for Task 04. Violation count: 0.

Table 25 reproduces all six feedback items. The feedback identifies the problem and its evidence, but does not expose the complete rubric or a reference answer.

No.	Rubric item	Judge feedback	Deduction
1	Core profitability metric	The 2022 gross margin was reported as 40.02%; the correct value is 41.00%, beyond the 0.5-percentage-point tolerance.	2
2	Profit-quality assessment	Only the cash collection ratio and net cash ratio were explicitly calculated. The report omitted core profit divided by total profit and its interpretation.	2
3	Profitability-driver decomposition	The report listed changes in gross margin and expense ratios but did not systematically attribute them, for example through a quantity–price decomposition.	2
4	Receivables analysis	The report omitted allowance sufficiency and customer concentration, and did not mark the missing receivables-aging note as unavailable.	2
5	Inventory analysis	The report omitted inventory-impairment analysis and the relation between inventory and revenue, and did not mark the missing inventory detail as unavailable.	2
6	Change attribution	The report verified the three-factor ROE product and gave a directional interpretation, but did not use chain substitution or an equivalent method to quantify each factor’s contribution.	2

Table 25: Complete textual feedback returned for Task 04.

The numerical and coverage implications of the feedback are shown in Table 26. These values are computed directly from the Stage 1 report and make explicit what the next execution should verify or add.

Feedback target	Evidence in the submitted report	Required correction or extension
Gross-margin arithmetic	2022 revenue was 115.05 and cost was 67.88, but the table reported 40.02%.	Recalculate $(115.05 - 67.88)/115.05 = 41.00\%$.
Core-profit share	Core profit was 14.89, 12.27, and 9.82; total profit was 16.00, 13.25, and 10.67 for 2023–2021.	Report 93.06%, 92.60%, and 92.03%, respectively, and interpret their stability.
Margin drivers	The report observed gross margin of 40.00%, 40.02%, and 42.00% and rising selling-expense ratios.	Separate price, volume, product-mix, and cost effects when supported; otherwise state which decomposition inputs are unavailable.
Receivables coverage	Turnover days were 34.8, 36.0, and 37.2 versus an industry value of 65 days.	Add allowance-sufficiency and customer-concentration checks and label the unavailable aging note.
Inventory coverage	Turnover days were 44.5, 44.4, and 46.8 versus an industry value of 120 days.	Add impairment adequacy; compare inventory growth with revenue growth (16.67% vs. 20.0% in 2023; 20.0% vs. 22.0% in 2022); label unavailable detail.
ROE attribution	The report verified the identity and gave only the direction of the three drivers.	A consistent 2021–2023 chain substitution gives +0.38, +0.59, and +1.30 percentage points from net margin, turnover, and leverage; the sum, +2.26 points, equals the ROE change.

Table 26: Detailed evidence and corrections corresponding to the six Stage 2 feedback items.

C.3 Stage 3 Memory Update

Stage 3 resumed the same conversation used to generate the report. The agent converted the six feedback items into persistent, task-general guidance rather than storing company-specific figures or conclusions. The memory update retained the prior guidance and inserted the new lessons at six locations: the profitability block, the asset-quality block, the DuPont block, two repeated application summaries, and the memory descriptor.

Memory Before and After Task 04	
Before	Existing guidance already covered stepwise core-profit arithmetic, ROIC consistency, missing-data labels, DuPont identity checks, cash-flow-pattern checks, goodwill and construction-in-progress analysis, statement reconciliation, and qualitative ratings.
Added after feedback	(1) Recalculate gross margin exactly; (2) always compute all three profit-quality ratios, including core profit/total profit; (3) use a systematic quantity–price or equivalent attribution for gross-margin changes; (4) assess receivables allowances and customer concentration and label unavailable aging data; (5) assess inventory impairment and inventory–revenue matching and label unavailable detail; and (6) quantify ROE-factor contributions with chain substitution.
Updated application summary <i>Check percentage calculations explicitly; cover all three profit-quality ratios; perform systematic gross-margin attribution; complete allowance, concentration, impairment, and matching checks; label unavailable notes; and quantify each ROE driver rather than reporting direction alone.</i>	

Figure 8: The Task 04 memory update. The retained content abstracts from the current company and records reusable failure-prevention guidance.

C.4 Stage 3 Skill Update

The skill update translated the same feedback into executable additions to the financial-analysis procedure. Nine change blocks modified the calculation checklist, driver-decomposition procedure, asset-quality checklist, DuPont procedure, three deduction lists and summary tables, and the final report-quality checklist. Table 27 shows the operational before–after differences.

Procedure block	Before Task 04	Added after Task 04
Calculation validation	Validate core-profit arithmetic and reconcile the result with operating profit.	Recalculate gross margin as $(\text{revenue} - \text{cost})/\text{revenue}$ and explicitly compute the cash collection ratio, net cash ratio, and core profit/total profit.
Gross-margin attribution	Discuss raw-material prices, product mix, pricing power, competition, and cost pass-through.	Require a systematic decomposition, including quantity and price contributions when supported, rather than a list of directional changes.
Receivables	Compare receivables growth, revenue growth, aging, allowance practice, and related-party collection periods.	Require allowance-sufficiency and top-customer-concentration analyses; mark unavailable aging-note data explicitly.
Inventory	Compare growth and turnover with revenue and industry values; include inventory-impairment analysis.	Require an explicit inventory–revenue matching test, impairment sufficiency, and an unavailable-data label when inventory details are absent.
ROE attribution	Check that the three-factor product equals ROE and that the contributions sum to the total change.	Require numerical chain-substitution contributions for net margin, asset turnover, and the equity multiplier; a product check and directional conclusion alone are insufficient.
Section-level deduction lists	List prior calculation, completeness, and attribution failure modes.	Add the six Task 04 failure modes and their two-point deductions under profitability, asset quality, and DuPont analysis.
Consolidated quality check-list	Check data completeness, accounting consistency, cash flow, financial risk, reconciliation, conclusions, and reporting quality.	Add explicit checks for percentage arithmetic, all three profit-quality ratios, systematic margin attribution, receivables and inventory coverage, and quantitative ROE attribution.

Table 27: Operational changes made to the reusable financial-analysis skill after Task 04.

C.5 Selected Final Persisted State

The preceding tables describe the edits. This subsection instead shows selected content as it appears in the persistent state after reflection. The excerpts are translated from the archived memory and skill files, with formatting normalized for the appendix. They include only the blocks materially changed by Task 04.

Final Memory Entry The feedback memory retained earlier lessons and incorporated the new findings into the existing profitability, asset-quality, and DuPont blocks. The final entry is identified as `financial-analysis-report-standards`; the selected post-update text is reproduced below.

Final Memory Entry — Selected Post-Update Content	
Profitability calculations	Gross margin must be recomputed as $(\text{revenue} - \text{cost})/\text{revenue}$. Percentage values must be checked explicitly rather than inferred from a trend table.
Profit-quality coverage	Explicitly calculate all three measures: cash collection ratio, net cash conversion ratio, and core profit/total profit. Missing one measure leaves the earnings-quality assessment incomplete.
Profitability attribution	A change in gross margin requires systematic attribution. Separate price, volume, product mix, raw-material cost, and cost pass-through effects when the inputs permit; otherwise mark the unavailable inputs.
Receivables	Cover turnover efficiency, allowance sufficiency, and customer concentration. If the aging note or customer detail is absent, label it unavailable rather than silently omitting the analysis.
Inventory	Cover turnover efficiency, impairment allowance, and inventory growth versus revenue growth. If inventory detail is absent, label it unavailable.
ROE attribution	Use chain substitution or an equivalent method to quantify the contributions of net margin, asset turnover, and the equity multiplier. A product check and directional explanation alone are insufficient.

Figure 9: Selected final content of the feedback memory after Task 04.

Final Skill: Calculation and Coverage Rules The skill incorporated the feedback at the point of execution rather than keeping it only as a retrospective note. The first affected block now requires the agent to complete the following checks while drafting the profitability and asset-quality sections.

Final Skill Excerpt 1 — Profitability and Asset Quality	
Calculation validation	For each year, recompute gross margin from revenue and cost; reconcile core profit with operating profit; and verify that summary-table values equal the detailed calculations.
Profit-quality metrics	Calculate cash collection ratio = cash collected/revenue, net cash conversion = operating cash flow/net profit, and core-profit share = core profit/total profit. Interpret all three.
Gross-margin drivers	Do not stop at year-to-year changes. Quantify price, volume, mix, raw-material cost, and cost pass-through effects when supported, and assess whether the identified drivers are sustainable.
Receivables checklist	(1) Turnover or aging; (2) allowance adequacy and industry comparison; (3) top-customer concentration; (4) related-party collection; and (5) an explicit unavailable-data label for a missing aging note.
Inventory checklist	(1) Inventory growth versus revenue growth; (2) turnover versus the industry; (3) impairment allowance and adequacy; and (4) an explicit unavailable-data label for missing inventory detail.

Figure 10: Selected final profitability and asset-quality instructions in the updated skill.

Final Skill: Quantitative ROE Attribution The final DuPont block specifies the calculation rather than merely requiring “deeper attribution.” Let m_t , u_t , and e_t denote net margin, asset turnover, and the equity multiplier in period t . For a base period 0 and current period 1, the skill now contains the following chain-substitution procedure:

Final Skill Excerpt 2 — Chain-Substitution Procedure	
$\Delta_{\text{margin}} = (m_1 - m_0)u_0e_0,$ $\Delta_{\text{turnover}} = m_1(u_1 - u_0)e_0,$ $\Delta_{\text{leverage}} = m_1u_1(e_1 - e_0),$ $\Delta\text{ROE} = \Delta_{\text{margin}} + \Delta_{\text{turnover}} + \Delta_{\text{leverage}}.$	
Required validation. Compute each contribution numerically, verify that their sum equals the observed ROE change, and recheck the inputs if the identity does not close. Report both the contribution values and their business interpretation.	

Figure 11: Final chain-substitution instructions added to the updated skill.

Final Skill: Embedded Failure Catalogue The updated skill also embeds the six lessons in its section-level failure catalogue. This makes the feedback available during both execution and final checking, rather than leaving it only in the task history.

Skill section	Final failure condition added after Task 04	Deduction
Profitability	Gross-margin error exceeds 0.5 percentage points.	2 points
Profitability	Any of the three profit-quality indicators is missing, including core-profit share.	2 points
Profitability	Gross-margin attribution remains a numerical comparison without systematic driver analysis.	2 points
Asset quality	Receivables omit allowance sufficiency, customer concentration, or an unavailable-data label for missing aging information.	2 points
Asset quality	Inventory omits impairment adequacy, inventory–revenue matching, or an unavailable-data label for missing detail.	2 points
DuPont	ROE change is described qualitatively without numerical chain-substitution contributions.	2 points

Table 28: Selected final failure conditions embedded in the updated skill.

Final Skill: Pre-Delivery Checks The final quality-control block converts all six feedback items into reusable cross-task checks without generalizing company-specific facts or exposing the full rubric. Before saving a deliverable, the agent verifies every item in Table 29.

No.	Final post-update check
1	Recompute gross margin and other percentage indicators directly from their source values.
2	Confirm that the cash collection ratio, net cash conversion ratio, and core-profit share are all calculated and interpreted.
3	Confirm that gross-margin changes receive systematic driver attribution rather than a directional description alone.
4	Confirm that receivables cover allowance adequacy and customer concentration, with missing aging data labeled unavailable.
5	Confirm that inventory covers impairment adequacy and growth relative to revenue, with missing detail labeled unavailable.
6	Confirm that ROE contributions are numerical and sum exactly to the observed ROE change.

Table 29: Selected final pre-delivery checks added after Task 04.