DATACLAWBENCH: AN AGENT BENCHMARK FOR EXPLORATORY REAL-WORLD FINANCIAL DATA ANALYSIS

Anonymous authors
Paper under double-blind review

ABSTRACT

Realistic analysis often begins without knowing where the relevant evidence lies, raising a fundamental question for autonomous data analysis: can agents discover and use the required evidence with limited human guidance? However, existing benchmarks typically evaluate such agents in human-guided settings, providing selected data sources, explicit data schemas, or cleaned data, thereby understating the exploratory burden. To evaluate autonomous exploratory analysis in a more realistic setting, we introduce DataClawBench, a benchmark built from financial think tank consulting scenarios where agents must independently explore unfamiliar, noisy, cross-domain data and produce conclusions. DataClawBench provides a unified real-world data environment containing 2.06 million records across enterprise, industry, and policy domains while preserving native data imperfections. On this environment, it defines 492 multi-step cross-domain tasks that withhold data source hints, complete schema documentation, and explicit noise descriptions. To reveal exploratory progress and failure modes beyond final outcomes, we annotate critical milestones and introduce a process-oriented evaluation framework. A systematic evaluation of ten advanced LLMs under the OpenClaw agent reveals that exploratory data analysis breaks agent reliability: more exploration does not reliably translate into task-relevant progress or correct final answers[1].

1 INTRODUCTION

The rapid advancement of end-to-end agents driven by large language models (LLMs) (Shen et al., 2025; Steinberger, 2025; Qiao et al., 2026) is reshaping data analysis. Traditionally, automated data analysis has often been formulated as static question answering, where a model directly produces an answer from specified text or tables. However, many complex real-world analytical tasks are exploratory before they are computational. Analysts often start with a concrete question, but the relevant evidence is not fully specified. They must inspect unfamiliar data environments, identify useful sources, align information across sources, handle data noise, and synthesize a conclusion (Kandel et al., 2012; Crisan et al., 2021). As LLM agents become more autonomous, evaluating their ability to handle such exploratory tasks is increasingly important.

Existing benchmarks have made substantial progress in evaluating data analysis capabilities. However, these are not fully aligned with the requirements of exploratory real-world data analysis. Many benchmarks (Huang et al., 2024; Egg et al., 2025; Wang et al., 2025; Bigeard et al., 2025; Ma et al., 2026) evaluate agents in human-guided data settings, where relevant sources, schemas, or data noise conditions are often partly specified. This reduces the need for agents to discover evidence, align information across sources, and handle data noise. Yet in complex real-world analyses, such data guidance is often unavailable or incomplete. Preparing the input can already solve part of the task. As a result, they may overestimate how reliably agents can operate in real-world exploratory settings.

To fill this gap in evaluation, we need a realistic setting where evidence relevant to the task is distributed across multiple data sources, with incomplete schema documentation and native noise not fully known in advance. Financial think tank consulting offers such a setting. In practice, analysts

[1] https://anonymous.4open.science/r/DataClawBench-556A

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Benchmark	No Source Guidance	No Schema Guidance	No Noise Guidance	Cross Domain	Process Evaluation	Real-World Scenario
Spider 2.0 (Lei et al., 2025)	×	×	×	×	×	✓
InfiAgent (Hu et al., 2024)	×	×	×	×	×	×
DACode (Huang et al., 2024)	×	×	×	✓	×	✓
DABstep (Egg et al., 2025)	✓	✓	×	✓	×	✓
FDABench (Wang et al., 2025)	✓	✓	×	✓	×	×
FinanceBench (Bigeard et al., 2025)	✓	✓	×	×	×	✓
DataClawBench	✓	✓	✓	✓	✓	✓

Table 1: Comparison of representative benchmarks along key properties for exploratory data analysis. Noise guidance refers to either an explicit description of data noise or data cleaning performed before evaluation. ✓ denotes the presence of the corresponding property, and × denotes its absence.

are often asked to answer concrete consulting questions, such as enterprise diagnosis, industry comparison, and policy impact assessment, among others (OECD, 2015; Henke et al., 2016). These tasks require analysts to connect evidence across operational records, industry indicators, and policy documents, turning imperfect data into verifiable analytical conclusions (Chu et al., 2016; Hellerstein et al., 2018). This makes it a natural testbed for evaluating exploratory data analysis.

Motivated by this setting, we propose DataClawBench, a benchmark for exploratory real-world financial data analysis. DataClawBench places agents in a unified underexplored data environment containing approximately 2.06 million real-world records across enterprise, industry, and policy domains, with native data noise preserved. It includes 492 expert-designed multi-step reasoning tasks derived from financial think tank consulting scenarios. Each task is annotated with a unique final answer for objective evaluation, as well as intermediate milestones for diagnosing the agent's reasoning process. Table 1 compares DataClawBench with representative benchmarks along key properties for exploratory data analysis. Our contributions are summarized as follows.

- We introduce exploratory data analysis as a challenging setting for evaluating autonomous data analysis agents. This setting brings source discovery, schema interpretation, and native data noise handling into the evaluated workload, rather than assuming these challenges have already been resolved.
- We construct DataClawBench, a benchmark that instantiates this setting in financial think tank consulting. DataClawBench restores the exploratory burden often abstracted away in human-guided benchmarks.
- We conduct a systematic evaluation of ten advanced LLMs using the OpenClaw agent. Seven models achieve below 50% accuracy on DataClawBench. More exploration does not reliably translate into progress toward the task goal.

2 Evaluation Protocol for Exploratory Data Analysis

Exploratory data analysis requires agents to first identify task-relevant evidence from heterogeneous data sources before deriving analytical conclusions. Unlike settings where sources, schemas, or cleaned tables are pre-specified, this process involves both evidence discovery and downstream reasoning. We therefore evaluate agents under a limited-guidance setting, measuring not only final answer correctness but also execution efficiency and milestone-level exploratory progress.

2.1 Problem Setting.

We formalize each task as a tuple $\mathcal{T} = (q, \mathcal{D}, \mathcal{P}, \mathbf{s}, \mathbf{m}, a)$, where q is a natural-language analysis question and $\mathcal{D} = \{d_1, d_2, \ldots, d_K\}$ denotes a data environment with K sources. For each question, there exists a latent set of task-relevant evidence $\mathcal{R}_q \subseteq \mathcal{D}$, which is not directly exposed to the agent. $\mathcal{P}$ denotes task-level data guidance available to the agent, such as source hints, schema descriptions, or noise descriptions. DataClawBench instantiates a limited-guidance evaluation setting. In contrast to human-guided benchmarks that provide non-trivial guidance $\mathcal{P}$ to narrow the search space, DataClawBench withholds task-specific source hints, complete schema documentation, and explicit noise descriptions. Agents must therefore infer $\mathcal{R}_q$ by exploring $\mathcal{D}$.

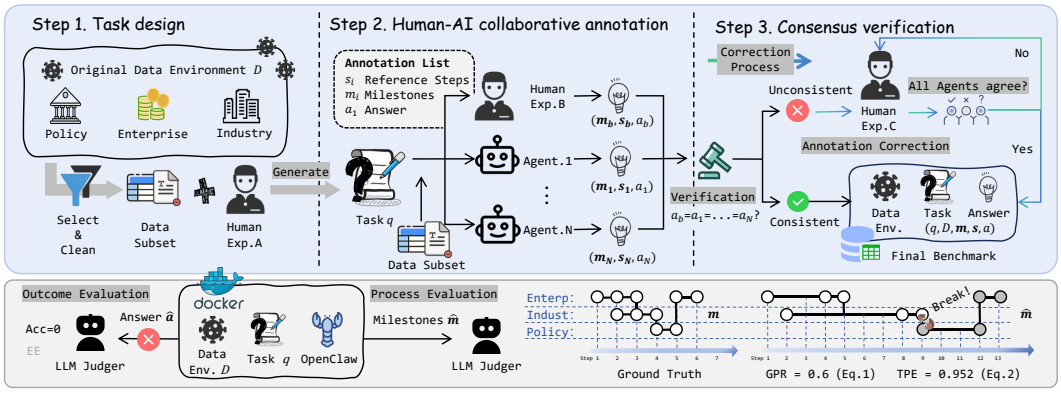


Figure 1: Overall framework of DataClawBench. **Top**. Data annotation pipeline. **Bottom**. Evaluation pipeline. Each agent runs in an isolated Docker container, locates relevant information in an underexplored data environment, performs numerical computation and text comprehension, and produces a final answer, which is then assessed by both outcome evaluation and process evaluation.

The remaining components support verifiable evaluation. $s = (s_1, \ldots, s_N)$ is an ordered reference reasoning sequence of N core analytical steps. These steps summarize the core solution logic rather than the full sequence of interactions required to execute the solution. $m = \{(k_j, v_j)\}_{j=1}^{M}$ is a set of M milestone key-value pairs representing critical intermediate results that must be established, either explicitly or implicitly, to reach the gold final answer a. An agent $\mathcal{A}$ receives $(q, \mathcal{D}, \mathcal{P})$ and produces $\mathcal{A}(q, \mathcal{D}, \mathcal{P}) = (\hat{s}, \hat{a})$, where $\hat{s}$ is the execution trajectory and $\hat{a}$ is the predicted answer.

2.2 Evaluation Metrics.

We evaluate agents along three complementary dimensions: effectiveness, efficiency, and exploratory progress.

Effectiveness. Effectiveness measures whether the agent reaches the correct conclusion. We use answer correctness (**Acc**), evaluated by an LLM judge that compares the predicted answer $\hat{a}$ with the gold answer a. For tasks containing L sub-questions, the judge scores each independently and reports the normalized accuracy in $[0, 1]$. The judging prompt is provided in Appendix E.

Efficiency. Efficiency measures whether agent reaches a correct conclusion with limited redundant exploration. For successful tasks, we define execution efficiency (**EE**) as $\eta = \frac{N}{T}$, where $N = |\mathbf{s}|$ is the reference trajectory length and $T = |\hat{\mathbf{s}}|$ is the number of agent execution steps. Following prior agent benchmarks (Xi et al., 2026b; He et al., 2026), we define an agent step as one model inference round and its tool observations, with multiple tool calls in the same round counted as one step.

Exploratory progress. Exploratory progress measures how much task-relevant progress an agent makes when the final answer is incorrect. We evaluate it using two milestone-based metrics. **Goal Progress Rate (GPR)** measures the fraction of reference milestones achieved during execution:

$$\mathrm{GPR} = \frac{1}{M} \sum_{j=1}^{M} \mathbb{I}(m_j), \tag{1}$$

where M is the number of gold milestones, and $\mathbb{I}(m_j) = 1$ if milestone m_j is correctly achieved in the predicted trajectory $\hat{\mathbf{s}}$. Milestone achievement is judged by an LLM based on either direct evidence in the trajectory or indirect evidence from correctly completed downstream milestones.

Temporal Progress Efficiency (TPE) further captures whether achieved milestones are reached in a timely manner:

$$\mathrm{TPE} = \frac{\sum_{j=1}^{M} \mathbb{I}(m_j) \cdot \gamma^{\max(t_j - N,\, 0)}}{\sum_{j=1}^{M} \mathbb{I}(m_j)} \in [0, 1], \tag{2}$$

where t_j is the first step at which milestone m_j is achieved, and $\gamma \in (0, 1]$ is the temporal decay factor. Milestones achieved within the first N steps receive full credit, while later milestones are exponentially discounted. Thus, $\mathrm{TPE} = 1$ indicates that all achieved milestones are reached on time, whereas lower values indicate delayed progress.

3 DATACLAWBENCH

3.1 UNDEREXPLORED REAL-WORLD DATA ENVIRONMENT

Data source composition. DataClawBench consists of a real-world data environment and analysis tasks. We first construct an underexplored real-world data environment $\mathcal{D}$ with approximately 2.06 million records from seven core data sources spanning enterprise, industry, and policy domains. Table 2 summarizes the source composition and data types, while Appendix Table 23 provides more details. The enterprise domain includes publicly listed technology-oriented companies and expert-selected key unlisted technology firms. The collected data cover static company profiles, annual financial and operating indicators, and textual descriptions of core competitiveness. The industry domain captures national- and provincial-level statistics on industrial economic performance, scale, and patent activity. The policy domain consists of policy texts and publication metadata on science and technology innovation at national and local levels.

Domain	Source	Type	Scale
Enterprise	Company profiles	Struct.	1,820K
	Company operations	Struct.	
	Core competitiveness	Unstruct.	
Industry	National statistics	Struct.	234K
	Regional statistics	Struct.	
Policy	Policy documents	Mixed	9K
	Release statistics	Struct.	
Internal knowledge base		Unstruct.	25

Table 2: DataClawBench data environment. Seven core data sources span three domains, expanding into 18 tables. Scale is measured in number of records.

Data characteristics. These sources form a cross-domain data network connected by implicit business attributes, such as registration location and industry classification. This structure enables the benchmark to link micro-level firm operations, meso-level industry patterns, and macro-level policy environments. Rather than curating the data into a simplified benchmark setting, DataClawBench deliberately retains real-world data imperfections, such as missing values, inconsistent units, ambiguous entity names, and multilingual expressions. Moreover, data schemas are not explicitly provided across sources. Some can be partially inferred from table headers, whereas others are difficult to recover due to irregular table structures.

Data provenance and anonymization. The core data of DataClawBench originates from the publishing team's long-term think-tank research and consulting practice, rather than synthetic samples or textbook examples. To preserve reasoning value while protecting commercial privacy, DataClawBench employs a three-stage anonymization pipeline. Full details are provided in Appendix D.2

3.2 TASK DESIGN AND CONSTRUCTION

Task composition. DataClawBench consists of 492 data analysis tasks designed by domain experts. As shown in Table 3, these tasks are organized into seven categories according to thematic scope. Most tasks involve joint analysis of micro-level enterprise performance, meso-level industry trends, and macro-level policy contexts, while international comparison tasks additionally require web searches for external information. Moreover, the tasks are designed to require multi-step reasoning, where intermediate results are often used as inputs to subsequent steps. Each task has a unique answer, supporting deterministic evaluation. Because of the required domain knowledge and annotation complexity, a human expert may spend more than one hour annotating a single task.

Difficulty annotation. We adopt different difficulty annotation strategies for numerical and text-intensive tasks. For the first four categories, which mainly involve numerical computation, difficulty is based on the number of annotated core reasoning steps: 2–3 steps for easy, 4–5 for medium, and

Category	Example Question	E / M / H	Total	Core
Enterprise–Industry Analysis	*In 2022, is the operating revenue of Company X higher than the total operating revenue of its industry in its province?*	115 / 111 / –	226	2–5
Enterprise–Industry–Policy Analysis	*Compare the number of central-level policies for Company A's industry with local policies for Company B's industry.*	10 / 66 / –	76	3–5
Comprehensive Decision	*In 2022, an automotive manufacturer scored each province's industrial supporting capacity before selecting a plant site. The scoring rules are as follows...what is the composite index value of the province with the highest industrial supporting composite index?*	6 / 45 / 19	70	2–8
International Comparison	*What is the ratio of Futu Holdings' net profit per employee to the median of China's capital market services industry?*	– / 25 / 14	39	4–7
Hypothesis Verification	*Verify whether enterprise total assets correlate with invention patent count in Consumer Electronics, restricted to provinces with above-average R&D density.*	– / 14 / 15	29	5–9
Industry Planning	*For Guangdong's consumer electronics industry, compare high-end and export-oriented strategies using inter-provincial ranking scores.*	– / 14 / 14	28	5–9
Risk Assessment	*If all national R&D tax incentives are cancelled, which three manufacturing industries suffer the largest average net profit margin decline?*	– / 11 / 13	24	5–9
Total		**131 / 286 / 75**	**492**	**2–9**

Table 3: Task distribution. Core denotes annotated core reasoning steps, not full agent execution steps. Example questions are abbreviated or partial due to space constraints.

6 or more for hard. For the last three categories, which involve long policy, difficulty is manually assigned by domain experts because step count alone cannot capture the reading and reasoning burden. A short core reference solution does not imply a short agent trajectory. The annotated core steps capture only the compressed reasoning structure, while actual execution requires additional exploration and correction. In practice, many agent trajectories extend to dozens of steps.

Human-in-the-loop annotation pipeline. As illustrated in Figure 1, DataClawBench adopts a three-stage human-in-the-loop pipeline, with expert task design, human-AI collaborative annotation, and consensus verification. This combines domain expertise with model-assisted efficiency.

❶ **Task design.** Domain Expert A first constructs a cleaned and idealized version of the data environment, and then designs analysis tasks based on it. ❷ **Human-AI collaborative annotation.** In the cleaned environment, Domain Expert B and multiple data analysis agents independently solve the same task. Each participant produces a candidate answer a, a reasoning trace $\mathbf{s}$, and milestone annotations $\mathbf{m}$. These parallel solutions provide references for subsequent verification. ❸ **Consensus verification.** Domain Expert C compares the human and agent outputs. If all participants reach the same final answer, the human annotation is directly accepted. Otherwise, Expert C examines the conflicting reasoning paths, corrects the annotation when necessary, and iterates the process until a consensus annotation is obtained.

This consensus-based verification reduces the risk of relying on a single annotator or model. To keep annotation feasible, the process is conducted in the cleaned data environment, so the final annotations focus on core solution steps rather than exploratory trial-and-error. We also release the cleaned data environment as a reusable resource for further research and benchmark development.

4 Experiments

We evaluate ten agents driven by LLMs spanning different model families and scales. All experiments use the OpenClaw agent framework. Each task is executed in an isolated Docker container with the full data environment $\mathcal{D}$ mounted as a read-only workspace. Agents may use any tools available in OpenClaw, including Python execution, file reading, web fetch, and web search, to explore the data and derive answers. A per-task timeout of 1200 seconds is enforced to bound computation. All evaluation judging is performed by GLM-5. Further details are provided in Appendix A.

Model	Easy		Medium		Δ Acc.	Hard		Δ Acc.	Overall	
	Acc. (↑)	EE (↑)	Acc. (↑)	EE (↑)	E→M	Acc. (↑)	EE (↑)	M→H	Acc. (↑)	EE (↑)
Claude Opus 5	**89.3**	0.37	**65.9**	0.42	23.4	40.6	0.45	25.4	**68.3**	0.40
Claude Opus 4.6	76.8	0.41	62.2	0.43	14.6	**44.8**	0.44	17.4	63.4	0.42
GPT-5.6-SOL	79.4	**0.52**	55.9	**0.55**	23.5	27.6	**0.66**	28.3	57.8	**0.54**
Gemini 3.1 Pro†	67.6	0.42	44.6	0.26	23.0	12.3	0.30	32.3	45.8	0.32
Minimax M2.7	62.6	0.31	37.2	0.21	25.4	19.3	0.26	17.9	41.3	0.26
Qwen3.5-Plus	45.0	0.31	40.0	0.20	5.0	16.2	0.29	23.8	37.7	0.24
DeepSeek-V3.2	45.0	0.15	35.8	0.15	9.2	8.9	0.20	26.9	34.1	0.16
GLM-5	45.8	0.32	33.0	0.32	12.8	12.6	0.30	20.5	33.3	0.32
Kimi-K2.5	41.2	0.37	22.6	0.31	18.6	14.0	0.26	8.6	26.3	0.33
GPT-5.4	23.1	0.45	26.5	0.46	−3.4	12.0	0.59	14.5	23.4	0.46
Mean	57.6	0.36	42.4	0.33	15.2	20.8	0.38	21.5	43.1	0.35

Table 4: Outcome scores. Arrows indicate the preferred direction of each metric. The Δ columns report adjacent difficulty-stage accuracy drops. †Gemini 3.1 Pro denotes Gemini 3.1 Pro Preview.

Model	Easy		Δ	Medium		Δ	Hard		Δ	Overall	
	GPR (↑)	TPE (↑)	Rank	GPR (↑)	TPE (↑)	Rank	GPR (↑)	TPE (↑)	Rank	GPR	TPE
Claude Opus 5	27.6	0.64	−1	**54.4**	0.59	0	**51.7**	0.54	1	**51.4**	0.57
Claude Opus 4.6	**42.9**	0.56	2	45.6	0.63	0	45.4	0.50	−2	45.1	0.59
GPT-5.6-SOL	17.3	**0.73**	−1	41.7	**0.80**	0	45.9	**0.84**	1	39.8	**0.81**
Gemini 3.1 Pro†	17.2	0.54	0	40.1	0.43	0	28.7	0.26	4	33.6	0.41
Minimax M2.7	16.3	0.40	−1	39.9	0.37	1	26.6	0.36	−1	33.2	0.37
Qwen3.5-Plus	11.7	0.29	0	24.9	0.37	−2	24.0	0.42	−1	21.7	0.37
DeepSeek-V3.2	16.4	0.36	3	28.7	0.33	1	22.4	0.34	3	24.7	0.34
GLM-5	7.3	0.38	−3	23.8	0.52	−1	20.0	0.54	−1	19.5	0.52
Kimi-K2.5	9.5	0.59	1	23.7	0.56	0	15.0	0.61	−4	19.2	0.57
GPT-5.4	6.0	0.68	0	24.2	0.76	1	18.8	0.73	0	18.5	0.75
Mean	17.2	0.52	—	34.7	0.54	—	29.8	0.51	—	30.7	0.53

Table 5: Process scores on incorrectly answered tasks with $\gamma = 0.9$. ΔRank measures the rank divergence between final accuracy and progress, computed as the Acc rank minus the GPR rank.

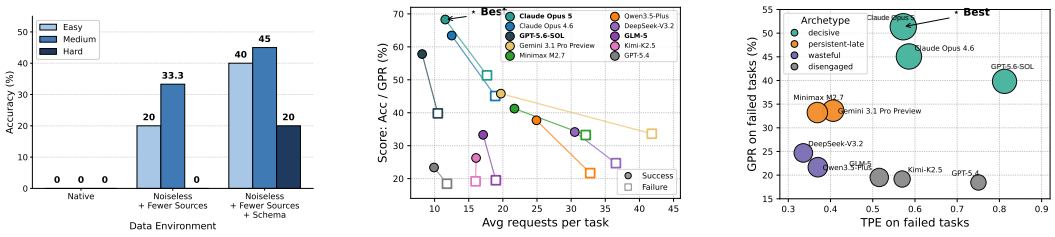
4.1 Main Results

Q1. Can current agents remain reliable when they must find the relevant data by themselves?

Not reliably. Table 4 summarises outcome scores across all 492 tasks. Even the strongest agent, Claude Opus 5, reaches only 68.3% overall accuracy. Seven of the ten models fall below 50%, and five fall below 40%. **No model approaches saturation, indicating that exploratory analysis over real-world data environments remains far from solved**. Performance also deteriorates sharply as task complexity increases. Averaged across the ten models, accuracy drops by 15.2 percentage points from Easy to Medium, but by 21.5 percentage points from Medium to Hard, a decline roughly 1.4× larger. Claude Opus 5 drops from 89.3% on Easy tasks to 40.6% on Hard tasks, while other models show even steeper declines, with the largest Acc drop reaching 55.3 percentage points. Per-category results for all ten models are provided in Appendix B.1.

Q2. Is longer exploration actually converted into useful progress?

Only partially. The key to exploration is not how long it lasts, but whether it translates into task progress. As shown in Figure 2b, failed tasks with longer exploration generally achieve higher GPR, indicating that additional requests can recover some intermediate progress. The early-stopping models, including GPT-5.4, Kimi-K2.5, and GLM-5, use only about 12–19 requests on failed tasks and reach roughly 18–20% GPR. By contrast, Gemini 3.1 Pro, Minimax M2.7, DeepSeek-V3.2, and Qwen3.5-Plus explore longer, using about 32–42 requests, and achieve higher GPR of around 22–34%. However, more requests do not translate into progress uniformly. Claude Opus 5 reaches 51.4% GPR with 17.7 requests, and GPT-5.6-SOL reaches 39.8% GPR with 10.5 requests, whereas Gemini and Minimax spend substantially more requests but obtain lower progress. **This suggests that exploration length alone is a weak proxy for useful progress. What matters is whether exploration remains goal-directed rather than longer**.



(a) Acc under different environments.

(b) Avg requests on failed tasks vs. GPR.

(c) TPE vs. GPR.

Figure 2: Three diagnostic views of agent behaviour on DataClawBench. (c) The ten models partition into four exploration archetypes, and outcome accuracy (bubble size) co-varies with this partition. Color = archetype.

Q3. What failure modes are hidden behind the same wrong answer?

The same wrong answer can hide very different process failure modes. Final accuracy only tells us whether an answer is correct. It does not show how the agent failed. The gap between Acc rank and GPR rank shows this clearly. Table 5 reports that on Hard tasks, Gemini 3.1 Pro ranks eighth by final accuracy but fourth by GPR. This means that many of its failed runs still reach useful milestones. Kimi-K2.5 shows the opposite pattern. It ranks sixth by accuracy but last by GPR, which means that its failed runs often reach fewer milestones. From the outcome view, these two agents may look comparable. From the process view, they are fundamentally different agents.

Q4. When agents fail, what exploration profiles do they exhibit?

Combining Acc, GPR, and TPE reveals four exploration profiles. In Figure 2c, the decisive solvers (Claude Opus 5, Claude Opus 4.6, and GPT-5.6-SOL) achieve high GPR without sacrificing TPE, reaching a substantial fraction of milestones early. They also has high Acc. The persistent but late archetype (Gemini 3.1 Pro Preview, Minimax M2.7) holds high GPR but low TPE, accumulating milestones eventually rather than early. The wasteful trial-and-error archetype (DeepSeek-V3.2, Qwen3.5-Plus) has moderate GPR and low TPE, spending many steps without proportionate progress. The disengaged archetype (GPT-5.4, Kimi-K2.5, GLM-5) has the lowest GPR but high TPE, they terminate early, so the few milestones they reach come early.

4.2 Failure Attribution

Q5. What makes exploratory data analysis difficult: noisy data or missing guidance?

Both matter. We re-evaluate Qwen3.5-Plus on originally failed tasks under three data environments with decreasing environmental uncertainty, as shown in Figure 2a. The first is the original real-world environment. The second removes data noise and irrelevant sources. The third further provides complete schema guidance on top of the second environment. Reducing environmental uncertainty improves accuracy, but the gains are uneven across difficulty levels. In the original environment, model fails on all sampled tasks. After removing noise and irrelevant sources, accuracy recovers to 20% on Easy and 33.3% on Medium, but remains 0% on Hard tasks. When complete schema guidance is added, accuracy further rises to 40% on Easy, 45% on Medium, and 20% on Hard tasks.

This shows that exploratory data analysis is not hard only because the data is noisy. Removing noise and irrelevant sources helps agents enter a more useful evidence space, especially on Easy and Medium tasks. However, Hard remain difficult until stronger schema guidance is provided, and even then recover only partially. This suggests that agents struggle with multiple forms of uncertainty at once: noisy evidence, irrelevant sources, and incomplete schema understanding. **Data preparation itself already performs part of the problem solving**. Other models show consistent trends, detailed in Appendix B.3.

Q6. Where do agents first lose the analytical thread?

Agents usually lose the analytical thread early, but stronger agents fail later than weaker ones. We localize the origin of each failed run by recording the first unachieved gold milestone m_k. Fig-

First-failing operation	Wrong-Answer Stop	Voluntary Give-up	Silent Stop	Timeout Kill	Other
Entity Attribute Lookup	37.2	45.7	10.3	5.3	1.4
Aggregate Count or Sum	69.0	10.6	10.6	8.8	1.0
Statistical Summary	56.8	22.2	9.0	11.5	0.4
Policy Lookup and Count	66.0	16.9	5.0	10.6	1.6
Comparison or Boolean Judgment	63.1	15.5	5.3	13.1	2.9
Ranking and Selection	56.9	25.0	7.5	6.9	3.8

Table 6: Termination behavior by first-failing operation for the eight representative agents. Category definitions are provided in Appendix B.2.

ure 3 reports this distribution for eight representative agents, grouped by difficulty. Failures are concentrated near the beginning of the analytical chain, but the depth of failure differs across models. Claude Opus 4.6, the strongest agent, is less likely to collapse at the very first milestone. For most agents, the first milestone M_1 accounts for a large share of failures. This pattern suggests that many agents derail before establishing the initial evidence base, framing the problem, or setting up intermediate variables. Stronger agents can often move beyond the initial evidence-acquisition stage and fail later.

Q7. Why do agents fail to recognize when they have lost the thread?

Whether an agent gives up or commits to a wrong answer depends strongly on whether the breakpoint produces an observable failure signal. Table 6 shows a sharp divide between explicit lookup failures and silent semantic failures. For Entity Attribute Lookup, Voluntary Give-up is the largest termination mode at 45.7%, suggesting that agents can often recognize when a concrete row, field, or entity is missing. By contrast, Wrong-Answer Stop dominates Aggregate Count or Sum, Policy Lookup and Count, and Comparison or Boolean Judgment at 69.0%, 66.0%, and 63.1%, respectively, while Ranking and Selection and Statistical Summary also exceed 56%. These operations can still return plausible outputs even when the underlying evidence is incomplete or misused, making the failure locally difficult to detect. This points to a planning-state miscalibration: agents are better at detecting missing evidence than recognizing when available evidence no longer supports the current plan. Once such a silent error occurs, later computations can remain internally coherent and reinforce the mistaken trajectory. Hard tasks contain more policy, aggregation, statistical, and ranking operations, creating more opportunities for locally plausible but globally invalid intermediate states. Thus, **agents lose the thread not only by making a wrong move, but by failing to recognize that the move has invalidated the plan**.

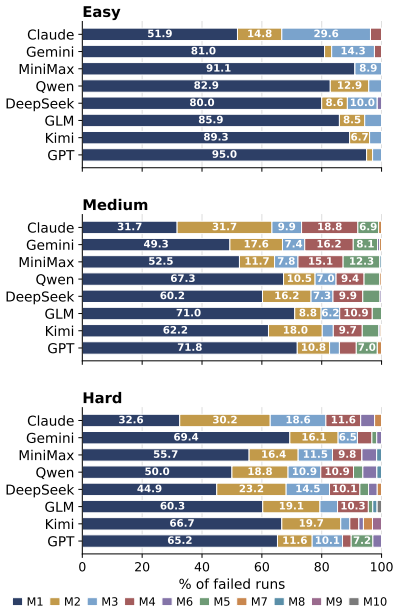


Figure 3: Position m_k of the first unachieved milestone, shown separately for Easy, Medium, and Hard tasks.

4.3 Reliability of the Evaluation Framework

We validate the reliability of our evaluation framework from three perspectives, including human alignment, cross judge consistency, and milestone robustness across reasoning paths.

Human alignment. We conduct a blinded human study on 12 tasks stratified by difficulty, covering eight representative agents, 96 trajectories, and 624 milestone decisions. Two annotators independently review the complete trajectories while blinded to agent identities and judge outputs. Table 8 reports Cohen's κ for final answer and milestone judgments, together with ± 1 step accuracy for first step localization. High human–human agreement indicates strong annotation reproducibility, while high judge–human agreement supports the reliability of the LLM judge. The lower first step score is expected because it additionally requires localizing the earliest divergence within a one step tolerance. Further details are provided in Appendix B.4.

Cross judge consistency. To rigorously address concerns about biases associated with model families, we evaluate whether model rankings remain stable when only the judge model changes, using Kendall's τ_b. Across GPT-5.6-SOL, Grok-4.6, and Hy4-preview, both final answer and GPR rankings remain largely consistent, as shown in Table 7. Human review links GPT-5.6-SOL's lower rank consistency to overly strict false negatives only for Claude Opus 4.6. This indicates that our main outcome and process conclusions are robust to judge choice.

Ranking	Hy4-preview	Grok-4.6	GPT-5.6-SOL
Acc	1.0	1.0	0.6
GPR	1.0	1.0	0.8
TPE	1.0	1.0	0.8

Table 7: Cross-judge consistency.

Comparison	Final-ans. $\kappa \uparrow$	Milestone $\kappa \uparrow$	First-step $\pm\mathbf{1} \uparrow$
Human–Human	0.9473	0.9036	0.8205
Judge–Human	0.8890	0.8154	0.6848

Table 8: human and judge agreement

Milestone robustness. We further test whether milestones generalize across reasoning paths through a blinded audit of 50 tasks across all difficulty levels, using trajectories from eight representative agents. Two domain experts independently identify the minimal indispensable semantic results C_t. Let G_t denote the annotated milestones. We define Milestone Coverage Rate (MCR) $\mathrm{MCR} = \frac{1}{T}\sum_{t=1}^{T} \mathbb{I}[C_t \subseteq G_t]$, which measures whether the annotated milestones cover this independently identified semantic core. MCR reaches 100%, 94%, and 94% on Easy, Medium, and Hard tasks, respectively, and 96% overall. This indicates that the milestones largely capture semantic states shared across reasoning paths rather than a single reference sequence.

5 Related Work

Data Analysis under human-guided Data Settings. Early research focused on static question answering progressed from single-table QA (Pasupat & Liang, 2015; Yu et al., 2018) to cross-database text-to-SQL (Lei et al., 2025), and from multi-hop textual reasoning (Yang et al., 2018; Wang et al., 2024b) to hybrid reasoning over tables and text (Chen et al., 2021; 2022). Recent work has begun to explore agent-oriented evaluation of multi-step data analysis. DA-Code (Huang et al., 2024) and InfiAgent-DABench (Hu et al., 2024) require agents to write code under explicitly stated data sources. FinanceBench (Bigeard et al., 2025) targets financial analysis over standardized SEC filings. More recently, DAB (Ma et al., 2026) evaluates heterogeneous enterprise databases. Despite these advances in task complexity and data scale, many of them still provide partial guidance about data sources, schemas, or noise conditions. Most of these benchmarks therefore primarily evaluate reasoning over human-guided data rather than analytical capability in underexplored environments.

Outcome-Oriented Evaluation Paradigms. Most agent benchmarks (Liu et al., 2024; Zhou et al., 2024; Chen et al., 2026) adopt an outcome-only evaluation approach, treating the reasoning process as a black box. Recent data analysis works (Egg et al., 2025; Gupta et al., 2026) explicitly identify process-level evaluation as an important future direction. A few recent works also have begun to explore process-level evaluation. AgentBoard (Ma et al., 2024) introduces a progress rate metric for incremental task completion, and Galileo (Galileo, 2025) proposes an action advancement measure to assess whether each step progresses toward the goal. Unlike the operation-step-level progress metrics above, DataClawBench evaluates goal achievement at the level of critical intermediate results, decoupling evaluation from specific execution paths to accommodate diverse analytical strategies while localizing semantic break points in the reasoning chain.

6 Conclusion

We present DataClawBench, a benchmark that evaluates LLM-driven agents on financial data analysis tasks in underexplored real-world data environments. It withholds source and schema guidance, preserves native data noise, and annotates tasks with process-level milestones. DataClawBench reveals not only whether an agent succeeds but how and where it fails. Experiments show that advanced LLMs struggle on DataClawBench and different models adopt markedly different exploration styles. Overall, DataClawBench provides a diagnostic testbed for probing the capability boundaries of autonomous financial data-analysis agents.

AI USE STATEMENT

We used generative AI to assist with dataset construction and agent evaluation, as well as manuscript editing for clarity and readability. We reviewed all AI-assisted work: human experts verified and adjudicated the benchmark annotations, judge reliability was assessed through human validation and cross-judge comparisons, and the authors reviewed the edited manuscript. The authors take responsibility for the final content.

REFERENCES

Alibaba Tianchi. A-share listed companies quarterly revenue prediction dataset. https://tianchi.aliyun.com/dataset/1074, 2023. Licensed under CC BY-NC-SA 4.0.

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

Jialong Chen, Xander Xu, Hu Wei, Chuan Chen, and Bing Zhao. Swe-ci: Evaluating agent capabilities in maintaining codebases via continuous integration. *arXiv preprint arXiv:2603.03823*, 2026.

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

Zhiyu Chen, Shiyang Li, Charese Smiley, Zhiqiang Ma, Sameena Shah, and William Yang 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*, pp. 6279–6292, 2022.

Xu Chu, Ihab F. Ilyas, Sanjay Krishnan, and Jiannan Wang. Data cleaning: Overview and emerging challenges. In *Proceedings of the 2016 International Conference on Management of Data (SIGMOD)*, pp. 2201–2206. ACM, 2016. doi: 10.1145/2882903.2912574.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021. doi: 10.48550/arXiv.2110.14168. URL https://arxiv.org/abs/2110.14168.

Anamaria Crisan, Brittany Fiore-Gartland, and Melanie Tory. Passing the data baton: A retrospective analysis on data science work and workers. *IEEE Transactions on Visualization and Computer Graphics*, 27(2):1860–1870, 2021. doi: 10.1109/TVCG.2020.3030340.

Alex Egg, Martin Iglesias Goyanes, Friso Kingma, Andreu Mora, Leandro von Werra, and Thomas Wolf. Dabstep: Data agent benchmark for multi-step reasoning. *arXiv preprint arXiv:2506.23719*, 2025.

Galileo. Introducing agentic evaluations, 2025. URL https://www.galileo.ai/blog/introducing-agentic-evaluations.

Nikita Gupta, Riju Chatterjee, Lukas Haas, Connie Tao, Andrew Wang, Chang Liu, Hidekazu Oiwa, Elena Gribovskaya, Jan Ackermann, John Blitzer, Sasha Goldshtein, and Dipanjan Das. Deepsearchqa: Bridging the comprehensiveness gap for deep research agents. *arXiv preprint arXiv:2601.20975*, 2026.

Pengfei He, Zhenwei Dai, Bing He, Hui Liu, Xianfeng Tang, Hanqing Lu, Juanhui Li, Jiayuan Ding, Subhabrata Mukherjee, Suhang Wang, et al. Traject-bench: a trajectory-aware benchmark for evaluating agentic tool use. In *International Conference on Learning Representations*, volume 2026, pp. 61766–61801, 2026.

Joseph M. Hellerstein, Jeffrey Heer, and Sean Kandel. Self-service data preparation: Research to practice. *IEEE Data Engineering Bulletin*, 41(2):23–34, 2018. URL http://sites.computer.org/debull/A18june/p23.pdf.

Nicolaus Henke, Jacques Bughin, Michael Chui, James Manyika, Tamim Saleh, Bill Wiseman, and Guru Sethupathy. The age of analytics: Competing in a data-driven world. Technical report, McKinsey Global Institute Research, 2016. URL https://www.mckinsey.com/capabilities/quantumblack/our-insights/the-age-of-analytics-competing-in-a-data-driven-world.

Xueyu Hu, Ziyu Zhao, Shuang Wei, Ziwei Chai, Qianli Ma, Guoyin Wang, Xuwu Wang, Jing Su, Jingjing Xu, Ming Zhu, Yao Cheng, Jianbo Yuan, Jiwei Li, Kun Kuang, yang Yang, Hongxia Yang, and Fei Wu. Infiagent-dabench: Evaluating agents on data analysis tasks. *Proceedings of Machine Learning Research*, 235:19544–19572, 2024.

Yiming Huang, Jianwen Luo, Yan Yu, Yitong Zhang, Fangyu Lei, Yifan Wei, Shizhu He, Lifu Huang, Xiao Liu, Jun Zhao, and Kang Liu. Da-code: Agent data science code generation benchmark for large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 13487–13521, 2024.

Liqiang Jing, Zhehui Huang, Xiaoyang Wang, Wenlin Yao, Wenhao Yu, Kaixin Ma, Hongming Zhang, Xinya Du, and Dong Yu. Dsbench: How far are data science agents from becoming data science experts? In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 32597–32649, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/50e9ad960ae78b741a6b4fea533f2eaf-Paper-Conference.pdf.

Sean Kandel, Andreas Paepcke, Joseph M. Hellerstein, and Jeffrey Heer. Enterprise data analysis and visualization: An interview study. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2917–2926, 2012. doi: 10.1109/TVCG.2012.219.

Fangyu Lei, Jixuan Chen, Yuxiao Ye, Ruisheng Cao, Dongchan Shin, Hongjin SU, Zhaoqing Suo, Hongcheng Gao, Wenjing Hu, Pengcheng Yin, Victor Zhong, Caiming Xiong, Ruoxi Sun, Qian Liu, Sida Wang, and Tao Yu. Spider 2.0: Evaluating language models on real-world enterprise text-to-sql workflows. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 28691–28735, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/46c10f6c8ea5aa6f267bcdabcb123f97-Paper-Conference.pdf.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 39578–39601, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/aca97732e30bcf1303bc22ac3924fd16-Paper-Conference.pdf.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, Minlie Huang, Yuxiao Dong, and Jie Tang. Agentbench: Evaluating llms as agents. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 52989–53046, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/e9df36b21ff4ee211a8b71ee8b7e9f57-Paper-Conference.pdf.

Xiaoqian Liu, Ke Wang, Yuchuan Wu, Fei Huang, Yongbin Li, Jianbin Jiao, and Junge Zhang. Agentic reinforcement learning with implicit step rewards. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ooROvpmxMV.

Chang Ma, Junlei Zhang, Zhihao Zhu, Cheng Yang, Yujiu Yang, Yaohui Jin, Zhenzhong Lan, Lingpeng Kong, and Junxian He. Agentboard: An analytical evaluation board of

multi-turn llm agents. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang (eds.), *Advances in Neural Information Processing Systems*, volume 37, pp. 74325–74362. Curran Associates, Inc., 2024. doi: 10.52202/079017-2365. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/877b40688e330a0e2a3fc24084208dfa-Paper-Datasets_and_Benchmarks_Track.pdf.

Ruiying Ma, Shreya Shankar, Ruiqi Chen, Yiming Lin, Sepanta Zeighami, Rajoshi Ghosh, Abhinav Gupta, Anushrut Gupta, Tanmai Gopal, and Aditya G Parameswaran. Can ai agents answer your data questions? a benchmark for data agents. *arXiv preprint arXiv:2603.20576*, 2026.

OECD. *Data-Driven Innovation: Big Data for Growth and Well-Being*. OECD Publishing, Paris, 2015. doi: 10.1787/9789264229358-en.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F. Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744. Curran Associates, Inc., 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/b1efde53be364a73914f58805a001731-Abstract.html.

Panupong Pasupat and Percy Liang. Compositional semantic parsing on semi-structured tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pp. 1470–1480, 2015.

Shuofei Qiao, Yanqiu Zhao, Zhisong Qiu, Xiaobin Wang, Jintian Zhang, Zhao Bin, Ningyu Zhang, Yong Jiang, Pengjun Xie, Fei Huang, et al. Scaling generalist data-analytic agents. In *International Conference on Learning Representations*, volume 2026, pp. 14942–14971, 2026.

Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. Rewarding progress: Scaling automated process verifiers for llm reasoning. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 60808–60838, 2025. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/98711dea460bdefe0e651ca23ec98ba2-Paper-Conference.pdf.

Minjie Shen, Yanshu Li, Lulu Chen, Zhichao Fan, Yanhang Li, and Qikai Yang. From mind to machine: The rise of manus ai as a fully autonomous digital agent. *arXiv preprint arXiv:2505.02024*, 2025.

Peter Steinberger. OpenClaw: An open-source autonomous AI agent, 2025. URL https://github.com/openclaw/openclaw.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. Math-shepherd: Verify and reinforce LLMs step-by-step without human annotations. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 9426–9439, Bangkok, Thailand, August 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.510. URL https://aclanthology.org/2024.acl-long.510/.

Shuai Wang, Ekaterina Khramtsova, Shengyao Zhuang, and Guido Zuccon. Feb4rag: Evaluating federated search in the context of retrieval augmented generation. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '24, pp. 763–773, New York, NY, USA, 2024b. Association for Computing Machinery. ISBN 9798400704314. doi: 10.1145/3626772.3657853. URL https://doi.org/10.1145/3626772.3657853.

Xingyao Wang, Yangyi Chen, Lifan Yuan, Yizhe Zhang, Yunzhu Li, Hao Peng, and Heng Ji. Executable code actions elicit better LLM agents. In *Proceedings of the 41st International Conference*

on Machine Learning, volume 235 of *Proceedings of Machine Learning Research*, pp. 50208–50232. PMLR, 2024c. URL https://proceedings.mlr.press/v235/wang24h.html.

Ziting Wang, Shize Zhang, Haitao Yuan, Jinwei Zhu, Shifu Li, Wei Dong, and Gao Cong. Fdabench: A benchmark for data agents on analytical queries over heterogeneous data. *arXiv preprint arXiv:2509.02473*, 2025.

Zhiheng Xi, Chenyang Liao, Guanyu Li, Zhihao Zhang, Wenxiang Chen, Binghai Wang, Senjie Jin, Yuhao Zhou, Jian Guan, Wei Wu, Tao Ji, Tao Gui, Qi Zhang, and Xuanjing Huang. AgentPRM: Process reward models for LLM agents via step-wise promise and progress. In *Proceedings of the ACM Web Conference 2026*, pp. 4184–4195. ACM, 2026a. doi: 10.1145/3774904.3792551. URL https://doi.org/10.1145/3774904.3792551.

Zhiheng Xi, Dingwen Yang, Jiaqi Liu, Jixuan Huang, Honglin Guo, Baodai Huang, Tinggang Chen, Qi Zhang, Zhonghang Lu, Chenyu Liu, et al. Agentgym2: Benchmarking large language model agents in de-idealized real-world environments. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 44451–44479, 2026b.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 2369–2380, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259. URL https://aclanthology.org/D18-1259/.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://iclr.cc/virtual/2023/oral/12647.

Tao Yu, Rui Zhang, Kai Yang, Michihiro Yasunaga, Dongxu Wang, Zifan Li, James Ma, Irene Li, Qingning Yao, Shanelle Roman, Zilin Zhang, and Dragomir Radev. Spider: A large-scale human-labeled dataset for complex and cross-domain semantic parsing and text-to-SQL task. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii (eds.), *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pp. 3911–3921, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1425. URL https://aclanthology.org/D18-1425/.

Kaiyan Zhang, Jiayuan Zhang, Haoxin Li, Xuekai Zhu, Ermo Hua, Xingtai Lv, Ning Ding, Biqing Qi, and Bowen Zhou. Openprm: Building open-domain process-based reward models with preference trees. In Y. Yue, A. Garg, N. Peng, F. Sha, and R. Yu (eds.), *International Conference on Learning Representations*, volume 2025, pp. 80846–80868, 2025a. URL https://proceedings.iclr.cc/paper_files/paper/2025/file/c919a2b5ec1de69f2629f9119676e336-Paper-Conference.pdf.

Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. The lessons of developing process reward models in mathematical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 10495–10516, Vienna, Austria, July 2025b. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-acl.547. URL https://aclanthology.org/2025.findings-acl.547/.

Congming Zheng, Jiachen Zhu, Zhuoying Ou, Yuxiang Chen, Kangning Zhang, Rong Shan, Zeyu Zheng, Mengyue Yang, Jianghao Lin, Yong Yu, and Weinan Zhang. A survey of process reward models: From outcome signals to process supervisions for large language models. *arXiv preprint arXiv:2510.08049*, 2025. doi: 10.48550/arXiv.2510.08049. URL https://arxiv.org/abs/2510.08049.

Jialun Zhong, Wei Shen, Yanzeng Li, Songyang Gao, Hua Lu, Yicheng Chen, Yang Zhang, Wei Zhou, Jinjie Gu, and Lei Zou. A comprehensive survey of reward models: Taxonomy, applications, challenges, and future. *arXiv preprint arXiv:2504.12328*, 2025. doi: 10.48550/arXiv.2504.12328. URL https://arxiv.org/abs/2504.12328.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. Webarena: A realistic web environment for building autonomous agents. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun (eds.), *International Conference on Learning Representations*, volume 2024, pp. 15585–15606, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/4410c0711e9154a7a2d26f9b3816d1ef-Paper-Conference.pdf.

Appendix Contents

A Experimental Details **15**

B Supplementary Results **15**

B.1 Cross-Task and Cross-Difficulty Results 15

B.2 Operation and Termination Taxonomies 19

B.3 Environmental Impact Analysis 19

B.4 Human–Judge Alignment 20

B.5 Cost Analysis 20

B.6 Cross-Benchmark Accuracy Illustration 21

C Detailed Comparison with Dense Process Reward Models **22**

D Supplementary Data and Task Details **22**

D.1 Data Environment Details 22

D.2 Data Anonymization Details 24

D.3 Data Copyright 24

D.4 Task Construction Details 25

D.4.1 Data Sources and De-identification Preprocessing 25

D.4.2 Task Extraction and Difficulty Stratification Strategy 25

D.4.3 Expert Recruitment and Annotation Guideline Development 25

D.4.4 Quality Control and Inter-Annotator Agreement 26

E Detailed Prompts **26**

F Case Study **29**

G Limitations and Future work **33**

A Experimental Details

For every evaluated model we call its native vendor API with default setting `temperature`, `top_p`, or `max_tokens`. We likewise use each vendor's default reasoning or thinking behavior, summarised in Table 9. Each agent is allocated its model's maximum supported context window, the OpenClaw orchestration layer does not impose an additional setting. The OpenClaw framework, Docker images, judge prompt, and per-task wall-clock timeout (1200 s) are kept identical across models and difficulty buckets, as stated in §4.

B Supplementary Results

B.1 Cross-Task and Cross-Difficulty Results

Figure 4 and Tables 10–19 report per-model accuracy broken down by task category and difficulty level. Rows correspond to task categories and columns to difficulty levels. A "–" entry indicates that the corresponding category has no tasks at the given difficulty.

Model	Context	Thinking Mode
Claude Opus 4.6	1M	high effort
Claude Opus 5	1M	medium effort
Gemini 3.1 Pro Preview	1M	high effort
GPT-5.4	1.05M	off
GPT-5.6-SOL	1.05M	medium effort
GLM-5	203K	thinking
DeepSeek-V3.2	131K	thinking
Minimax M2.7	197K	thinking
Qwen3.5-Plus	1M	thinking
Kimi-K2.5	262K	thinking

Table 9: model evaluation configuration. "Thinking Mode" lists each model's vendor-API default for its reasoning / extended-thinking mode at the time of evaluation.

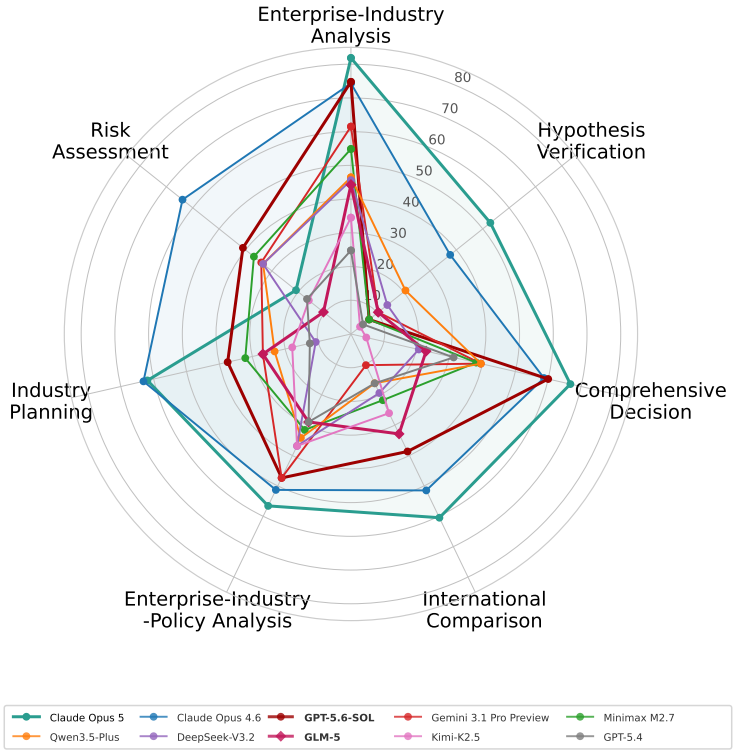


Figure 4: Accuracy by task category across all models.

Category	Easy	Medium	Hard	Overall
Enterprise–Industry Analysis	76.5	72.1	–	74.3
Enterprise–Industry–Policy Analysis	80.0	47.0	–	51.3
Comprehensive Decision	76.7	65.6	36.8	58.7
International Comparison	–	52.3	50.0	51.5
Hypothesis Verification	–	50.0	26.1	37.6
Industry Planning	–	71.4	54.8	63.1
Risk Assessment	–	66.7	61.5	63.9
Overall	76.8	62.2	44.8	63.4

Table 10: Accuracy (%) of Claude Opus 4.6 by task category and difficulty.

Category	Easy	Medium	Hard	Overall
Enterprise–Industry Analysis	43.5	49.5	–	46.5
Enterprise–Industry–Policy Analysis	70.0	28.8	–	34.2
Comprehensive Decision	33.3	50.0	16.7	39.5
International Comparison	–	15.3	17.9	16.2
Hypothesis Verification	–	42.9	0.0	20.7
Industry Planning	–	35.7	10.7	23.2
Risk Assessment	–	27.3	38.5	33.3
Overall	45.0	40.0	16.2	37.7

Table 11: Accuracy (%) of Qwen3.5-Plus by task category and difficulty.

Category	Easy	Medium	Hard	Overall
Enterprise–Industry Analysis	45.2	45.9	–	45.6
Enterprise–Industry–Policy Analysis	60.0	33.3	–	36.8
Comprehensive Decision	16.7	28.4	3.5	20.6
International Comparison	–	22.3	14.3	19.4
Hypothesis Verification	–	28.6	0.0	13.8
Industry Planning	–	21.4	0.0	10.7
Risk Assessment	–	36.4	30.8	33.3
Overall	45.0	35.8	8.9	34.1

Table 12: Accuracy (%) of DeepSeek-V3.2 by task category and difficulty.

Category	Easy	Medium	Hard	Overall
Enterprise–Industry Analysis	43.5	45.0	–	44.2
Enterprise–Industry–Policy Analysis	80.0	21.2	–	28.9
Comprehensive Decision	33.3	28.9	5.3	22.9
International Comparison	–	35.6	28.0	32.9
Hypothesis Verification	–	14.3	6.7	10.3
Industry Planning	–	35.7	17.9	26.8
Risk Assessment	–	13.6	7.7	10.4
Overall	45.8	33.0	12.6	33.3

Table 13: Accuracy (%) of GLM-5 by task category and difficulty.

Category	Easy	Medium	Hard	Overall
Enterprise–Industry Analysis	89.6	73.9	–	81.9
Enterprise–Industry–Policy Analysis	90.0	51.5	–	56.6
Comprehensive Decision	83.3	76.1	39.5	66.7
International Comparison	–	71.3	41.1	60.5
Hypothesis Verification	–	71.4	35.6	52.9
Industry Planning	–	67.9	56.0	61.9
Risk Assessment	–	9.1	30.8	20.8
Overall	89.3	65.9	40.6	68.3

Table 14: Accuracy (%) of Claude Opus 5 by task category and difficulty.

Category	Easy	Medium	Hard	Overall
Enterprise–Industry Analysis	80.9	68.5	–	74.8
Enterprise–Industry–Policy Analysis	70.0	43.9	–	47.4
Comprehensive Decision	66.7	73.3	26.3	60.0
International Comparison	–	41.7	33.3	38.7
Hypothesis Verification	–	7.1	6.7	6.9
Industry Planning	–	39.3	35.7	37.5
Risk Assessment	–	43.9	38.5	41.0
Overall	79.4	55.9	27.6	57.8

Table 15: Accuracy (%) of GPT-5.6-SOL by task category and difficulty.

Category	Easy	Medium	Hard	Overall
Enterprise–Industry Analysis	19.1	30.6	–	24.8
Enterprise–Industry–Policy Analysis	80.0	21.2	–	28.9
Comprehensive Decision	3.3	44.4	8.8	31.2
International Comparison	–	17.3	14.3	16.2
Hypothesis Verification	–	7.1	2.2	4.6
Industry Planning	–	10.7	14.3	12.5
Risk Assessment	–	9.1	23.1	16.7
Overall	23.1	26.5	12.0	23.4

Table 16: Accuracy (%) of GPT-5.4 by task category and difficulty.

Category	Easy	Medium	Hard	Overall
Enterprise–Industry Analysis	67.0	55.9	–	61.5
Enterprise–Industry–Policy Analysis	80.0	42.4	–	47.4
Comprehensive Decision	60.0	48.4	11.8	39.5
International Comparison	–	14.0	3.6	10.3
Hypothesis Verification	–	21.4	0.0	10.3
Industry Planning	–	42.9	10.7	26.8
Risk Assessment	–	28.8	38.5	34.0
Overall	67.6	44.6	12.3	45.8

Table 17: Accuracy (%) of Gemini 3.1 Pro Preview by task category and difficulty.

Category	Easy	Medium	Hard	Overall
Enterprise–Industry Analysis	65.2	44.1	–	54.9
Enterprise–Industry–Policy Analysis	40.0	30.3	–	31.6
Comprehensive Decision	50.0	45.6	16.7	38.1
International Comparison	–	28.0	10.7	21.8
Hypothesis Verification	–	14.3	0.0	6.9
Industry Planning	–	39.3	25.0	32.1
Risk Assessment	–	22.7	48.7	36.8
Overall	62.6	37.2	19.3	41.3

Table 18: Accuracy (%) of Minimax M2.7 by task category and difficulty.

Category	Easy	Medium	Hard	Overall
Enterprise–Industry Analysis	41.7	27.0	–	34.5
Enterprise–Industry–Policy Analysis	60.0	33.3	–	36.8
Comprehensive Decision	0.0	7.3	0.0	4.7
International Comparison	–	22.7	32.1	26.1
Hypothesis Verification	–	0.0	6.7	3.4
Industry Planning	–	21.4	14.3	17.9
Risk Assessment	–	7.6	23.1	16.0
Overall	41.2	22.6	14.0	26.3

Table 19: Accuracy (%) of Kimi-K2.5 by task category and difficulty.

B.2 Operation and Termination Taxonomies

Table 6 cross-tabulates two orthogonal taxonomies: what operation the agent was attempting when it first deviated from the gold milestone chain, and how the run ultimately terminated. We define each taxonomy below.

Operation taxonomy. We use a six-class operation taxonomy applied to the first un-achieved milestone of each failed run. The six classes are:

- **Entity Attribute Lookup**, mapping a single named entity to a single attribute, for example "Industry of *X* Company" or "Operating revenue of *X* in 2022".
- **Aggregate Count or Sum**, counting or summing across a filtered group, for example "Total chemical enterprises" or "Number of provinces with valid disclosure".
- **Statistical Summary**, computing a median, mean, a derived ratio such as per-capita or year-over-year change, or a composite normalized score.
- **Policy Lookup and Count**, querying or counting policy records by theme $\times$ issuer $\times$ region.
- **Comparison or Boolean Judgment**, performing pairwise comparison, threshold judgment, difference, or yes-or-no conclusion.
- **Ranking and Selection**, returning a top-k list, a rank position, or selection of the first, last, highest, or lowest item.

Termination taxonomy. Each failed run is assigned exactly one termination cause according to how the trajectory ends. We use five priority-ordered classes.

- **Timeout Kill** denotes runs externally terminated after exhausting the time budget.
- **Silent Stop** denotes runs that end without a usable final response. This often occurs when transient context saturation causes the model to stop reasoning.
- **Voluntary Give-up** denotes runs where the agent explicitly states that it cannot find the required evidence or that the requested information appears unavailable.
- **Wrong-Answer Stop** denotes runs where the agent emits a non-empty final answer and stops as if the task were solved, but the answer is incorrect.
- **Other** collects residual cases that do not fit the above categories.

B.3 Environmental Impact Analysis

To verify that the noise and reasoning decomposition reported in §4.2 is not specific to Qwen3.5-Plus, we replicate the paired rerun study on a second model under the same three progressively cleaned data environments. Figure 5 shows that GLM-5 reproduces the same directional pattern as Qwen3.5-Plus in Figure 2a, where cleaning improves Easy and Medium accuracy more than Hard accuracy. The phenomenon is therefore reproducible across models rather than an artifact of any single backbone.

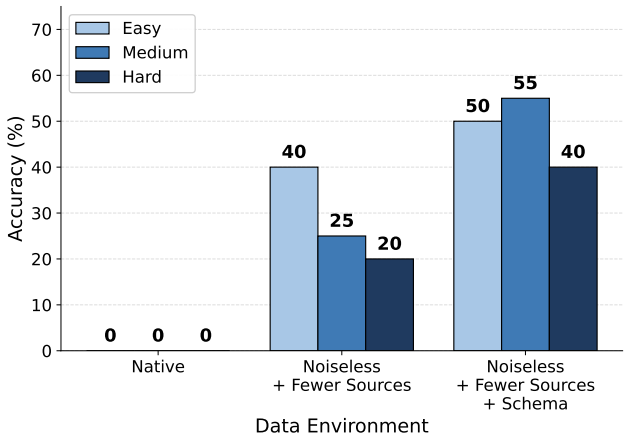


Figure 5: GLM-5 accuracy under progressively cleaned data environments.

B.4 Human–Judge Alignment

This appendix details the agreement study summarized in §4.3.

Annotation protocol. Two annotators independently label final-answer correctness, milestone achievement, and the first step at which each achieved milestone occurs. They use the same rubrics as the LLM judge and have no access to agent identities or LLM-judge outputs. Two annotators independently reviewed the complete trajectories while blinded to agent identities and judge outputs, and a third annotator resolved disagreements.

Judge–human failure modes. We manually reviewed all judge–human disagreements and identified four main failure modes: ❶ output-format mismatches despite numerically correct final answers. ❷ missed evidence caused by candidate-snippet filtering of long tool outputs. ❸ disagreement over the first step at which a milestone was achieved. and ❹ incorrect milestone judgments caused by reasoning beyond the available trajectory evidence.

B.5 Cost Analysis

Table 20 reports the computational cost of evaluating the DataClawBench benchmark for each model, priced at OpenRouter list rates recorded on each model's evaluation date (full unit prices in Table 21). The original models use the 2026-04-25 price snapshot, while Claude Opus 5 and GPT-5.6-SOL use the 2026-08-24 price snapshot. Per-task cost is the sum of input, output, cache-read, and cache-write token costs computed from each model's own per-million-token rates.

Cost patterns. The cost spread across models is governed by three distinct forces along the price-volume axis. At the high end, premium per-token rates dominate. Claude Opus 4.6 charges \$5 and \$25 per million input and output tokens with no cache-read reuse, accumulating roughly 35.5% of the total benchmark spend. In the middle band, volume and rate trade off. Qwen3.5-Plus processes the most input tokens per task at 1.84M but lands mid-pack in total cost because its per-million input rate is roughly twenty times lower than Opus. At the low end, cache-heavy pathways and cheap rates take over. Minimax M2.7 stays cheapest at roughly \$0.12 per task, a cost gap of more than an order of magnitude below Opus, supported by the largest cache-read volume at \$0.059 per million. GLM-5 follows at roughly \$0.19 per task, then GPT-5.6-SOL at roughly \$0.25 per task, while Kimi-K2.5 and Claude Opus 5 remain \$0.68 per task with substantial cache-read reuse.

Per-outcome request counts and exploration archetypes. The per-outcome request columns "Avg Reqs (C)" and "Avg Reqs (W)" supply the request-count evidence behind the four archetypes introduced in §4.1, walked through here in the same order. The decisive solvers Claude Opus 5, Claude Opus 4.6, and GPT-5.6-SOL issue 11.6, 12.5, and 8.2 requests per correctly answered task

and 17.7, 18.9, and 10.5 per failed task, respectively, the lowest combinations among models with high GPR. The persistent-but-late group of Gemini 3.1 Pro Preview and Minimax M2.7 sits in a similar range on correct tasks at 19.7 and 21.7 but expands sharply on failed tasks to 41.8 and 32.1, consistent with extended exploration that arrives at milestones late rather than early. The wasteful trial-and-error group of DeepSeek-V3.2 and Qwen3.5-Plus runs high on both columns, at 30.5 and 24.9 on correct tasks and 36.6 and 32.8 on failed tasks, spending many steps without proportionate progress. The disengaged group of GLM-5, GPT-5.4, and Kimi-K2.5 remains relatively short: GLM-5 uses 17.1 requests on correct tasks and 19.0 on failed tasks, while GPT-5.4 and Kimi-K2.5 stay at or below 16.1 on both columns.

Contrasting the decisive solver with the persistent-but-late group sharpens the cost-versus-progress tradeoff. Claude Opus 5 reaches 51.4% GPR with only 17.7 failed-task requests, and GPT-5.6-SOL reaches 39.8% GPR with 10.5 requests, whereas Gemini and Minimax issue 41.8 and 32.1 requests for lower GPR of 33.6% and 33.2%. Per failed-task request, Claude Opus 5 extracts roughly 3.6 times the milestone progress that Gemini does, mirroring the decisive solvers' higher TPE of 0.57 and 0.81 against Gemini's 0.41 and Minimax's 0.37.

Model	Tasks	Reqs	Total Tok	Avg Input	Avg Output	Avg Cache	Avg Reqs (C)	Avg Reqs (W)	Cost/Task	Total
Claude Opus 4.6	492	7,358	282M	569K	4.5K	0	12.5	18.9	$2.956	$1,454.26
Claude Opus 5	492	6,728	361M	48.9K	3.9K	680K	11.6	17.7	$0.681	$335.12
Qwen3.5-Plus	492	14,739	907M	1,840K	6.5K	0	24.9	32.8	$0.488	$240.01
DeepSeek-V3.2	492	17,033	700M	1,371K	9.9K	43K	30.5	36.6	$0.350	$172.17
GPT-5.4	492	5,610	207M	325K	2.3K	93K	9.9	11.8	$0.870	$428.21
GPT-5.6-SOL	492	4,523	190M	86.0K	2.2K	297K	8.2	10.5	$0.254	$124.82
GLM-5	492	9,046	408M	172K	6.0K	651K	17.1	19.0	$0.194	$95.37
Gemini 3.1 Pro	492	15,730	751M	926K	9.9K	590K	19.7	41.8	$2.089	$1,027.96
Minimax M2.7	492	13,806	553M	182K	12.4K	930K	21.7	32.1	$0.124	$61.15
Kimi-K2.5	492	7,883	465M	517K	2.3K	427K	16.1	16.0	$0.326	$160.25
Total	4,920	102,456	4,823M	–	–	–	–	–	–	**$4,099.33**

Table 20: Computational cost of evaluating DataClawBench, priced at OpenRouter list rates recorded on each model's evaluation date. "Reqs" is the total number of model API requests across those tasks. "Total Tok" is the total token volume. Per-task columns show averages over tasks. "Avg Reqs (C)" and "Avg Reqs (W)" are the average number of requests per task on correctly and incorrectly answered tasks, respectively. "Cost/Task" is the sum of input, output, cache-read, and cache-write costs at each model's OpenRouter per-million-token rate (see Table 21).

Model	Input	Output	Cache Read	Cache Write
Claude Opus 4.6	$5.00	$25.00	$0.50	$6.25
Claude Opus 5	$5.00	$25.00	$0.50	6.25
Qwen3.5-Plus	$0.26	$1.56	–	$0.325
DeepSeek-V3.2	$0.252	$0.378	$0.0252	–
GPT-5.4	$2.50	$15.00	$0.25	–
GPT-5.6-SOL	$2.00	$10.00	$0.20	2.5
GLM-5	$0.60	$2.08	$0.12	–
Gemini 3.1 Pro Preview	$2.00	$12.00	$0.20	$0.375
Minimax M2.7	$0.30	$1.20	$0.059	–
Kimi-K2.5	$0.44	$2.00	$0.22	–

Table 21: Per-million-token unit prices on OpenRouter (USD), using the price snapshot recorded on each model's evaluation date. "–" indicates the price tier is not advertised by the provider.

B.6 Cross-Benchmark Accuracy Illustration

Figure 6 provides an illustrative comparison of Qwen3.5-Plus accuracy across several recent data-analysis benchmarks. This comparison is not intended as a controlled ranking across benchmarks. Different benchmarks vary in task design, evaluation protocol, tool environment, and result source. Finance Agent and TaxEval results are from official leaderboards, while others are evaluated with OpenClaw. Therefore, the figure should be interpreted only as a coarse indication that DataClaw-Bench poses a challenging evaluation setting.

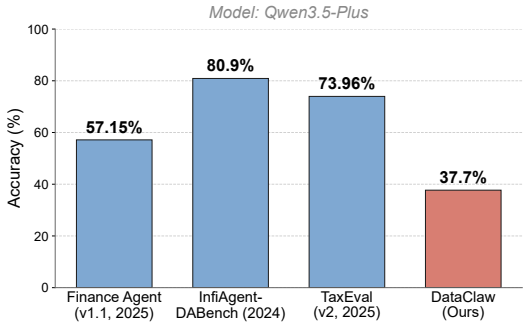


Figure 6: Accuracy across data analysis benchmarks.

C Detailed Comparison with Dense Process Reward Models

Our results show that real-world data analysis depends not only on final-answer correctness but also on whether the execution trajectory makes sustained progress through critical intermediate states. Current agent training relies primarily on final-answer supervision, which is insufficient for the open-ended exploration and multi-stage analysis these tasks require (Yao et al., 2023; Hu et al., 2024; Jing et al., 2025).

The milestone annotations and process scoring mechanism provided by DataClawBench make it possible to transform whether an agent reaches key intermediate states during the process into milestone-level reward signals (Ouyang et al., 2022). Based on these signals, one can train a reward model that lies between a sparse Outcome Reward Model (ORM) and a dense Process Reward Model (PRM). Compared with ORM signals (Zhong et al., 2025), these signals are denser and can help alleviate the supervision sparsity and credit assignment difficulties (Cobbe et al., 2021; Lightman et al., 2024). Compared with PRM signals, they are coarser but naturally aligned with the goal-oriented and path-diverse nature of real-world workflows. Thus, DataClawBench enables progress-based verifier training and trajectory-level reward shaping in the data-analysis setting.

More specifically, compared with dense PRM, which provides stepwise rewards (Zheng et al., 2025), milestone-based supervision can mitigate three issues. First, dense PRM often requires costly human annotation (Lightman et al., 2024), or relies on extensive sampling to construct training corpus (Wang et al., 2024a; Zhang et al., 2025a). Second, annotation methods that infer step-level values from final rewards are susceptible to sampling variance and to the quality of automatic annotation strategies (Wang et al., 2024a; Zhang et al., 2025a; Setlur et al., 2025; Zhang et al., 2025b; Liu et al., 2026). Third, exploratory retrieval and local backtracking are often necessary for information gathering and subsequent error correction. Dense PRM tends to mistakenly penalize such actions, whose short-term benefits may be unclear but which are indispensable for subsequent progress (Wang et al., 2024c; Xi et al., 2026a; Setlur et al., 2025).

D Supplementary Data and Task Details

D.1 Data Environment Details

The data environment is organized around three thematic domains of enterprise, industry, and policy. Subcategories cover enterprise profiles together with region-specific variants, enterprise core competitiveness, business status, regional and national industry statistics, policy releases, and full policy text. The taxonomy follows the structure of real research and consulting workflows. The environment contains 18 independent data files, where each file counts as one source. Of these, 17 reside under the three theme subdirectories, and one serves as an internal business-logic knowledge base unassigned to any theme domain.

Table 23 provides a complete inventory of all 18 tables in the DataClawBench data environment, including field counts, record counts, and representative fields. The majority of these tables are independently collected and curated by the author team, while a subset of the Enterprise and Industry tables additionally incorporates open-source curated financial data (Alibaba Tianchi, 2023).

Dimension	Value	Notes
Domains	3	Enterprise, industry, and policy. Counts only the 17 theme-domain files.
Core data sources	7	Three enterprise sources for profiles, core competitiveness, and business status. Two industry sources for regional and national statistics. Two policy sources for release status and full text. Per-file mapping appears in Table 23.
Total source files	18	Each file is treated as one source. All files are mounted into the container workspace at runtime. Composition is detailed in Table 23.
Format	Mainly CSV	Files are primarily CSV and contain both structured fields and long-form unstructured text.
Time span	Mainly 2022	Statistical periods vary across sources.
Data usability issues	–	Missing values, definition mismatches, inconsistent naming, and others, reflecting real-world data noise.

Table 22: Data Environment Summary

Domain	Source	Table	Records	Fields	Key Attributes
Enterprise	company profile	company_profile	7,295	29	bmCode, industry, province, listing date
		company_profile_as	1,282	29	Asia regional subset, identical schema
		company_profile_eu	1,201	29	Europe regional subset, identical schema
		company_profile_na	791	29	North America regional subset
		company_profile_oc	401	29	Oceania regional subset
	company operation	company_operation_status	353,438	9	Annual KPIs covering assets, revenue, profit, and R&D
		company_operation_status_*	781,772	5	Per-indicator detail rows for each enterprise-year
		company_operation_yearly_*	669,169	5	Year-over-year change metrics by enterprise
	company core	company_core	4,907	8	Core competitiveness narratives
Industry	nation-wide industry	national_industry_status	12,107	9	National industry aggregates
		national_industry_status_*	43,938	7	Granular per-indicator national breakdowns
		national_industry_yearly_*	776	7	Year-over-year national aggregates
	regional industry	regional_industry_status	170,896	10	Provincial industry aggregates
		regional_industry_status_*	3,173	8	Granular per-indicator provincial breakdowns
		regional_industry_yearly_*	3,207	8	Year-over-year provincial aggregates
Policy	policy resource	policy_resource	1,129	19	Full policy texts with metadata
	policy release	policy_release_status	7,792	9	Policy publication statistics
Internal	–	internal_metrics	25	2	Metric definitions (EN + ZH)

Table 23: Complete data environment inventory. **Domain** denotes the top-level analytical dimension under the Enterprise–Industry–Policy framework. **Source** indicates the thematic data source that groups related tables within a domain (e.g., *company profile* aggregates the global table and its four regional subsets). **Table** is the physical CSV/relational table name on disk. **Records** reports the number of rows. **Fields** reports the number of columns in the table schema. **Key Attributes** summarizes the representative columns or semantic role of each table.

D.2 Data Anonymization Details

To preserve the dataset's inferential value and cross-domain relational structure while protecting commercial privacy and meeting academic compliance requirements, DataClawBench applies a three-stage de-identification pipeline of identifier pseudonymization, distribution-preserving perturbation, and cross-domain consistency validation.

Identifier Pseudonymization. Key identifiers capable of directly revealing corporate identity are irreversibly pseudonymized. Stock codes are mapped to pseudo-identifiers through custom cryptographic hashing, removing any deterministic linkage to the original securities. Enterprise names are reconstructed under controlled stochastic substitution that respects industry-specific semantic features and conventional naming patterns, retaining contextual plausibility within each industry while blocking reverse identification.

Distribution-Preserving Perturbation. Quantitative operational metrics such as revenue, profit, R&D expenditure, and patent counts are perturbed with a least-significant-digit mechanism. The mechanism applies controlled random perturbation only to the low-order digits of each value, preserving overall magnitude and relative rankings while disrupting exact-match identification. Unlike global additive noise, the perturbation targets only the low-precision representation layer of the data, retaining the original statistical moments such as mean, variance, skewness, and kurtosis as well as industry quantile structure. Kolmogorov–Smirnov tests confirm that the cumulative distribution functions of sub-industry subsets show no significant deviation between pre- and post-perturbation samples ($p > 0.05$), preserving the statistical fidelity required for cross-enterprise comparison, trend extrapolation, and econometric modelling.

Cross-Domain Consistency Validation. To prevent perturbation-induced inconsistencies in industry-level aggregates, we re-derive industry aggregate indicators from the perturbed enterprise data and align them with the original industry-level statistics on aggregate structure, mean offset, and distribution shape. The procedure preserves the data noise inherent in real data ecosystems, including mismatches in statistical scope, financial rounding conventions, cross-period disclosure lags, and measurement errors. As a result, the de-identified dataset retains the relational structure and causal constraints required for cross-domain inference and knowledge graph construction.

Together, these stages eliminate individual re-identification risk while preserving the structural, statistical, and business-logic associations across domains required for analytical use.

D.3 Data Copyright

We clarify that the released benchmark is designed for non-commercial academic research and will include only data that are publicly available, redistribution-permitted, author-curated, or transformed into de-identified derived indicators. We do not release restricted proprietary databases or raw materials obtained under non-disclosure agreements.

As shown in Appendix D.1, most data are independently collected and curated by the author team. A subset of the Enterprise and Industry tables additionally incorporates open-source curated financial data (Alibaba Tianchi, 2023). (Alibaba Tianchi, 2023) is released under the CC BY-NC-SA 4.0 license, which permits non-commercial reuse, redistribution, and adaptation, provided that appropriate attribution is given and derivative materials are shared under the same or a compatible license. We will make the dataset license explicit and separate it from the code license: the code is released under MIT, while the data are released under source-specific non-commercial terms.

The author-collected data were accumulated through long-term research practice and curated from the following channels:

- Enterprise Data: Derived from publicly disclosed annual and financial reports of listed and key unlisted firms. For release, we retain de-identified factual indicators and derived statistics, and avoid redistributing restricted raw report materials or proprietary commercial compilations.
- Industry Data: Sourced through formal academic research collaboration with a partner think-tank. The collaboration operates under strict non-disclosure and academic-only use

agreements. Only de-identified aggregate indicators authorized for academic release are included.

- Policy Data: Collected from official government publications and open portals. We include official public documents and factual metadata where redistribution is legally permitted, and exclude internal drafts, restricted metadata, and materials without clear redistribution permission.

In addition, as shown in Appendix D.2, we apply a three-stage de-identification and transformation pipeline to Enterprise and Industry data. These stages preserve the statistical relationships needed for benchmark evaluation while reducing linkage to original commercial or institutional materials.

D.4 Task Construction Details

D.4.1 Data Sources and De-identification Preprocessing

The raw corpus of DataClawBench is not synthesized by large language models nor drawn from pedagogical examples. It originates from the publishing team's long-term research and consulting work on enterprise operations, industrial evolution, and public policy. We use 2022 multi-dimensional, industry-representative data as the foundation, organized into Minimal Business Logic Units, hereafter MBLUs, along an Enterprise, Industry, and Policy three-dimensional cross-analysis framework. During preprocessing, the team applies de-identification procedures that strip sensitive identifiers and potential model-knowledge-leakage risks while retaining the informational noise, missing values, and logical noise characteristic of real think-tank workflows. This design keeps the evaluation environment close to real business scenarios and avoids the inflated performance models can exhibit on idealized data.

D.4.2 Task Extraction and Difficulty Stratification Strategy

Based on the de-identified MBLUs, we extract high-frequency real-world problem sets from typical think-tank analytical workflows. Tasks are stratified into three difficulty tiers according to their cognitive complexity and cross-domain integration requirements.

- **Easy** tasks focus on single-point fact retrieval, structured information extraction, and basic policy clause matching.
- **Medium** tasks require multi-source information alignment, causal inference, or policy-to-enterprise impact chain analysis.
- **Hard** tasks involve cross-domain data analysis, long-text comprehension, factual reasoning, or decision-making under complex constraints.

D.4.3 Expert Recruitment and Annotation Guideline Development

Annotation and quality verification are undertaken by an interdisciplinary expert team jointly established by a research institute and a university. Each candidate must have at least three years of research or graduate-level study experience in industrial economics, public policy, or corporate strategy, be proficient in quantitative and qualitative analytical methods, and pass benchmark assessments covering domain knowledge, logical reasoning, and text comprehension.

Prior to formal annotation, the technical team compiled the DataClawBench Annotation Guidelines. The guide underwent three rounds of pilot annotation and iterative refinement involving both human experts and AI agent testing. The final version covers task boundary definitions, quantitative criteria for difficulty classification, answer structure specifications, evidence traceability requirements, typical positive and negative samples, AI agent consensus thresholds and validation protocols, and contingency procedures for common ambiguities. The guidelines were finalized after pre-experimental validation of operability and calibration of multi-agent consistency standards, providing a standardized framework for subsequent human-AI collaborative annotation.

D.4.4 Quality Control and Inter-Annotator Agreement

To ensure the academic rigor and business applicability of the dataset, we designed a closed-loop quality control workflow comprising five stages of pilot annotation, consistency assessment, guideline iteration, formal annotation, and multi-level verification.

1. For **Easy** difficulty, we use a pipeline combining domain knowledge graph construction, templated generation, and automated rule verification. After technical team review, 131 valid questions are retained.

2. For **Medium** and **Hard** difficulty, an initial pool of over 400 high-value questions is manually curated by expert teams from the research institute and university. After the technical staff deliver the annotation guidelines, the university team performs the annotation. Each sample undergoes **back-to-back double-blind annotation** by at least two independent annotators, followed by cross-verification by third-party senior researchers who did not participate in the initial annotation.

3. In **AI agent consensus verification**, each annotated sample is independently assessed by multiple AI agents against the annotation guidelines for rationality, evidentiary completeness, and domain validity. Only samples on which all agents reach unanimous agreement pass validation. Samples with divergent AI evaluations or failed validation are escalated to human experts for re-annotation. Controversial cases identified through this process are systematically analyzed and fed back into refining the guidelines.

4. In **final verification**, the technical team conducts item-by-item review of AI-validated results against the annotation specifications, excluding samples with logical discontinuities, missing evidence, or deviations from business scenarios. The procedure yields 286 valid medium-difficulty and 75 valid hard-difficulty QA pairs.

Notably, DataClawBench includes 39 web-dependent tasks in the International Comparison category, accounting for 7.9% of the 492 tasks. The information sought through web search in these tasks consists exclusively of historical facts with fixed, unambiguous answers. The information scope is restricted to retrieving specific historical financial reports (e.g., 2022 10-K and F-1 filings) of US-listed companies. The intended target source is the SEC EDGAR database. Unlike general web pages, the SEC EDGAR is a legally mandated, entirely public, and permanent document archive. It requires no proprietary access, and its historical records are not subject to deletion or unannounced modification. The gold answers are extracted directly from these permanent SEC filings, ensuring that the benchmark remains fully reproducible over time, regardless of how live web search indexes evolve.

SEC filings are the canonical evidence used to construct the gold answers, but the current evaluation does not impose a network-level SEC-only allowlist. This is intentional: agents are evaluated on whether they recover the correct evidence and conclusion, not on whether they follow one prescribed retrieval path. This choice is intentional and consistent with outcome-oriented evaluation in open-web research (Gupta et al., 2026). We do not guarantee run-level determinism for an autonomous agent using live web infrastructure.

E Detailed Prompts

This section reproduces the three prompt templates used by the DataClawBench evaluation pipeline. Placeholders in angle brackets are substituted at runtime from the task definition or from the agent's recorded trajectory.

Agent task prompt. Each agent under test receives a single user message assembled from the task's question, output guidelines, and a one-line statement of permitted data sources. Tasks in the International Comparison category additionally permit web search. All other categories restrict the agent to the local database mounted at `./database/`. The full template appears in the box below.

Agent task prompt template

```
<question>

Output guidelines:
<guidelines>

<data_sources>
```

The <data_sources> line is set to either "You may use files under ./database/ and web search." for International Comparison tasks, or "Only use files under ./database/." for all other categories.

Outcome judge prompt for Acc. When the agent terminates, its final answer is graded by an LLM judge against the gold answer. The judge receives the original task prompt for context, an Expected Behavior summary, the agent's final text, and a category-dependent rubric. Single-answer tasks use a binary match rubric. Multi-part answer tasks use a per-part vector rubric whose total is the average of part scores.

Outcome judge prompt template

```
You are a grading function. Your ONLY job is to output a single
JSON object.

CRITICAL RULES:
- Do NOT use any tools (no Read, Write, exec, or any other tool
calls)
- Do NOT create files or run commands
- Do NOT write any prose, explanation, or commentary outside the
JSON
- Respond with ONLY a JSON object | nothing else

Be a strict evaluator. Judge the final assistant message against
the task and rubric.

## Task
<task_prompt>

## Expected Behavior
<expected_behavior>

## Agent final answer
<agent_final_text>

## Grading Rubric
<rubric>

Score each criterion from 0.0 to 1.0.

Respond with ONLY this JSON structure (no markdown, no code fences,
no extra text):
{"scores": {"criterion_name": 0.0}, "total": 0.0, "notes": "brief
justification"}
```

Process judge prompt for GPR. The process judge evaluates which gold-defined milestones were achieved during the agent's trajectory. It receives the gold reference steps, the milestone list, and a step-indexed reconstruction of the agent's execution. Tool outputs are pre-filtered to "candidate" snippets containing numeric matches against milestone values, reducing prompt size while preserving the evidence the judge needs. TPE is computed deterministically from the GPR result and per-milestone first-occurrence step indices, requiring no additional LLM call.

Process judge prompt template for GPR

```
You are a milestone evaluator for an AI agent benchmark. Your task
is to
determine which milestones an agent achieved during its execution
trajectory.

## Reference Solution Path

The following ordered steps describe the gold-standard approach to
solving this
task. Milestones are key intermediate results that should be
produced along
this path.

### Gold Steps (in order)
<numbered gold steps>

### Milestones to Evaluate
<numbered M1..Mk milestones, each as "key: expected value">

## Agent Execution Trajectory

Below is a reconstruction of the agent's work, organized by step
(assistant
turn). For each step, "Assistant" is the agent's own text output.
"Candidate"
sections are excerpts from tool execution outputs where
milestone-relevant
values were detected by numeric matching. These candidates may or
may not be
true milestone achievements | you must verify the semantic context.

<per-step trajectory blocks>

## Evaluation Rules

1. Direct evidence: a milestone is achieved if the trajectory
clearly shows
   the agent computed or obtained the expected value (or within 1%
   relative
   error) in the CORRECT semantic context. A number appearing in an
   unrelated
   context does NOT count.

2. Temporal coupling inference: milestones follow a logical
dependency chain.
   If a downstream milestone is correctly achieved, its upstream
   dependencies
   can be inferred as achieved.

3. Chain-break identification: if the final answer is INCORRECT,
identify the earliest milestone in the logical chain that was NOT
achieved -- this is the "break point" where the agent's reasoning
diverged.

4. Different-but-valid paths: judge milestone achievement based on
whether
   the agent obtained the correct intermediate values, regardless
   of method.

## Output Format

Respond with ONLY a JSON object. No markdown fences, no extra text.
```

```
{"milestones": [{"key": "...", "achieved": true, "evidence_type":
"direct",
  "first_step": 4, "reason": "..."}, ...],
 "break_point": null,
 "chain_summary": "one-sentence summary"}
```

F CASE STUDY

We present four representative case studies that illustrate how process-oriented evaluation surfaces diagnostic signals invisible to outcome-only accuracy. The first three cases use Claude Opus 4.6, one of the strongest models on DataClawBench, and span a correct-but-inefficient trajectory, an incorrect trajectory with an early breakpoint, and an incorrect trajectory with a late breakpoint. The fourth case fixes one task and contrasts how five different models traverse it, surfacing the exploration archetypes from §4.1. For readability, each trajectory below preserves the model's text reasoning verbatim where it is illuminating and summarizes tool calls together with their outputs. Gray boxes contain the task specification, and colored boxes contain the model trajectory.

Case 1. Correct answer, inefficient path. This task, drawn from the International Comparison category at medium difficulty, asks for ZEEKR's 2022 asset turnover ratio and its gap from the median of China's automobile manufacturing industry. The reference trajectory is four steps. Claude Opus 4.6 reaches the correct answer in 29 API requests, a 7.25× step expansion.

Case 1 (Claude Opus 4.6, 29 requests, answer correct)

Question. Against the backdrop of intensive policy rollouts for the new energy vehicle industry, what was ZEEKR's asset turnover ratio in 2022? How did it compare to the median asset turnover ratio of China's automobile manufacturing industry as a whole?
Reference steps.

1. Extract ZEEKR 2022 net revenues (RMB 31,899,448 thousand) and total assets (RMB 19,477,316 thousand) from the ZEEKR F-1 prospectus; compute asset turnover = 1.64.
2. From `company_profile.csv`, filter enterprises with `industry = Automobile manufacturing`, yielding 230 enterprises.
3. From `company_operation_status.csv`, obtain revenue and total assets for the 230 enterprises; compute per-company asset turnover; compute industry median = 0.59.
4. ZEEKR minus industry median $= 1.64 - 0.59 = 1.05$.

Gold answer. `[1.64, 1.05]`.

Claude Opus 4.6 trajectory.

1. Opens with "I'll start by checking what's available in the database and searching for the relevant financial data," and fires a directory listing plus a web search for ZEEKR in parallel.
2. Runs `head -5` against six CSV schemas simultaneously, an over-broad schema probe.
3. Greps for ZEEKR name variants across three profile files. All empty. Observes: "The database uses anonymized company names."
4. Pivots to an industry-based search. The keyword for automobile manufacturing yields 242 matches. Also scans the overseas-listing profile files in case ZEEKR was listed abroad.
5. Tries the parent company Geely. Finds only a supplier record that mentions ZEEKR as a customer, with no financials.
6. Fires a web search for the ZEEKR F-1 prospectus. SEC returns 403. First failed external fetch.
7. In parallel, pulls automobile-industry medians from `national_industry_status_detail.csv`, obtaining median total assets and median operating revenue.
8. Recognizes that `median(revenue)/median(assets)` is not equivalent to `median(revenue/assets)` and commits to the per-company computation.
9. Runs a Python script over all 242 automobile-industry IDs to compute per-company ratios. First attempt returns an absurd median of 51,601.32 from a unit mismatch.

10. Diagnoses the bug: "The units are NOT consistent – different companies use different units." Values appear in yuan, 10K yuan, 100K yuan, 1M yuan, and 10M yuan.
11. Rewrites the script with a unit-conversion table, recomputes, and obtains industry median asset turnover = 0.59 across 230 companies.
12. Resumes the external hunt for ZEEKR's financials across additional sites. Two more web fetches fail or return garbled data.
13. Finally retrieves clean figures from a third-party analytics site. Total assets = RMB 19,477 M, revenue = RMB 31,899 M.
14. Computes $31{,}899/19{,}477 = 1.6378 \rightarrow 1.64$.
15. Runs a third Python verification that recomputes the industry median from scratch, then a fourth one-line recomputation of the final difference. Both are redundant.
16. Outputs `[1.64, 1.05]`.

Inefficiency pattern. Each individual step is defensible, yet the compound trajectory exhibits five distinct sources of slack. First, schema probing before narrowing to the relevant tables. Second, double verification of ZEEKR's absence in the local database via multiple grep variants. Third, a unit-handling bug loop requiring two Python iterations. Fourth, four web-fetch dead-ends before a usable source is found. Fifth, two redundant re-verification passes on arithmetic that had already settled. Outcome-only evaluation records this task as a clean success, and EE captures the 7.25× inflation relative to the gold path.

Case 2. Incorrect answer, early breakpoint on policy filtering. This task, drawn from the Comprehensive Decision category at hard difficulty, asks for the top province's weighted composite score in pharmaceutical manufacturing under a four-indicator scheme. The four indicators are enterprise agglomeration (weight 30%), R&D expenditure as a share of revenue (30%), regional policy coverage intensity (20%), and R&D human resource penetration rate (20%); policy coverage intensity is defined as each province's pharmaceutical-related policy count divided by the national total of pharmaceutical-related policies. The reference trajectory is eight milestones. Claude Opus 4.6 consumes 18 API requests and achieves 5 of 8 milestones, yet fails at M2 by counting all national policies (602) instead of the 80 pharmaceutical-related ones. The error propagates through the policy support indicator and inverts the final ranking, yielding 0.80 for Jiangsu instead of the gold answer 0.92 for Shanghai.

Case 2 (Claude Opus 4.6, 18 requests, answer incorrect)

Question. In 2022, a strategic consulting firm ranks provinces by their attractiveness in pharmaceutical manufacturing using a four-dimensional weighted scoring system. The four indicators, all min-max normalized then weighted, are enterprise agglomeration (30%), R&D expenditure as a share of revenue (30%), regional policy coverage intensity (20%), and R&D human resource penetration rate (20%). Policy coverage intensity is the ratio of relevant policy items in each province to the total number of relevant policies nationwide. What is the composite score of the top-ranked province?

Reference steps.

1. From `regional_industry_status.csv` filtered to pharmaceutical manufacturing, obtain the national enterprise total = 449.
2. From `policy_release_status.csv` filtered to pharmaceutical manufacturing, obtain the national pharmaceutical-related policy count = 80.
3. From `regional_industry_status.csv`, identify 16 provinces with complete pharmaceutical-manufacturing coverage.
4. Compute Shanghai industry agglomeration = 0.1203.
5. Compute Shanghai R&D intensity = 0.2548.
6. Compute Shanghai policy support = 0.1375.
7. Compute Shanghai talent density = 0.162.
8. Compute Shanghai composite score after min-max normalization = 0.916.

Gold answer. `0.92`.

Claude Opus 4.6 trajectory.

1. Lists the database directory and probes the schemas of seven candidate CSVs in parallel, isolating `regional_industry_status.csv` and `policy_release_status.csv` as the relevant aggregate sources.
2. Filters `regional_industry_status.csv` to pharmaceutical manufacturing and aggregates per-province enterprise totals. Reaches 449 nationwide across 16 valid provinces, achieving M1 and M3.
3. **Break point.** Reads `policy_release_status.csv` and finds a row with `targetName = total published policy count, value = 602` and an empty `industryId`. Adopts 602 as the national denominator for "relevant policies" without filtering on `industry = pharmaceutical manufacturing`. The correct denominator, 80, sits in the same file under `industryId = 1` but is never queried.
4. For each province, divides the provincial pharmaceutical policy count by 602 to compute the policy intensity ratio, locking in a denominator that is 7.5 times too large.
5. Spends the next several requests debating the R&D indicator. Catches an outlier-driven mean of 19960% in one province, oscillates between mean and median, briefly attempts company-level aggregation across `company_operation_status.csv`, then returns to the regional pre-aggregated mean R&D ratio. M5 and M7 both achieved with correct provincial values.
6. Recomputes the composite score with min-max normalization across provinces. Because the policy denominator is inflated, every province's policy contribution is uniformly compressed; provinces strong on policy support such as Shanghai lose ground and provinces strong on the other three axes such as Jiangsu move to the top.
7. Outputs **`0.80`** for Jiangsu, cross-checks with a mainland-only re-run that returns the same value, and confirms the answer. Incorrect.

Break-point analysis. The failure is a single missed filter at M2 in the Policy Lookup and Count subtask category. Its structural cost is disproportionate to its locality. The flawed national denominator propagates linearly into M6 and rescales the relative weight of the four indicators after normalization, flipping the winning province from Shanghai to Jiangsu. Yet the agent still achieves 5 of 8 milestones, including every retrieval, unit-handling, and per-indicator computation step that does not depend on the policy denominator. Outcome-only evaluation collapses this run into a single 0.0 identical to a model that never reached any milestone, while the chain-break decomposition isolates the single false step that controls the answer.

Case 3. Incorrect answer, late break point. This task, from the Enterprise–Industry Analysis category at medium difficulty, asks for the signed difference between the median operating profit of the Real Estate industry (containing company A) and that of the Financial Industry (containing company B). The reference trajectory is five steps. Claude Opus 4.6 achieves the first four milestones correctly but drops the sign on the final arithmetic, delivering an absolute value. Outcome-only scoring assigns 0.0, yet GPR shows the agent reaches 4/5 milestones before the break.

Case 3 (Claude Opus 4.6, 14 requests, answer incorrect)

Question. What is the difference between the median operating profit amount of the industry of Zhao Ye Hua Chang Real Estate Development Co., Ltd. and that of the industry of Tong Tong Ze Hong Securities Co., Ltd.?

Reference steps.

1. From `company_profile.csv`, obtain the industry of Zhao Ye Hua Chang Real Estate Development Co., Ltd. = Real Estate.
2. From `national_industry_status.csv`, obtain the Real Estate median operating profit = 130,368,786 yuan.
3. From `company_profile.csv`, obtain the industry of Tong Tong Ze Hong Securities Co., Ltd. = Financial Industry.
4. From `national_industry_status.csv`, obtain the Financial Industry median operating profit = 1,010,930,425 yuan.
5. Compute Real Estate minus Financial Industry $= 130{,}368{,}786 - 1{,}010{,}930{,}425 = -880{,}561{,}639.0$.

Gold answer. `-880561639.0`.

Claude Opus 4.6 trajectory.

1. Opens with "Let me start by exploring the database structure and finding information about these two companies." Lists the database directory and previews the schema of `company_profile.csv`.
2. Recognizes that the company names are pinyin transliterations of anonymized Chinese names and extracts the salient keywords Real Estate Development and Securities for substring matching.
3. Greps the Hua Chang Real Estate tag and finds five real-estate candidates. Only one carries the Zhao Ye prefix; Claude identifies it as the intended match.
4. Greps Tong Tong Ze Hong and locates the corresponding Securities firm. Both company-to-industry milestones cleanly achieved.
5. Records the mapping. Company A maps to Real Estate (industryId=4). Company B maps to Financial Industry (industryId=7).
6. Greps `national_industry_status.csv` for the median operating profit indicator in Real Estate. Retrieves 1303.68786 in units of 100,000 yuan.
7. Greps the same indicator for Financial Industry. Retrieves 101.0930425 in units of 10,000,000 yuan. Notes the mismatched units between the two rows.
8. Performs unit normalization. Real Estate: $1303.68786 \times 10^5 = 130{,}368{,}786$ yuan. Financial Industry: $101.0930425 \times 10^7 = 1{,}010{,}930{,}425$ yuan. Both conversions numerically correct.
9. **Break point.** Writes: "The difference $= |1{,}010{,}930{,}425 - 130{,}368{,}786| = 880{,}561{,}639$ yuan." Silently wraps the subtraction in absolute-value bars and reorders the operands.
10. Briefly second-guesses the unit conversion rather than the sign, pivots to re-expressing everything in a common base unit, then stops without revisiting the arithmetic framing.
11. Outputs **`880561639.0`**. Incorrect, off by a sign.

Break-point analysis. All four retrieval-and-normalization milestones are clean. The failure is a single absolute-value reflex applied to a signed quantity that the question explicitly defines as A minus B. Because the sign error occurs at the terminal milestone, outcome-only evaluation penalizes the task identically to Case 2, despite roughly 97% of the reasoning chain being correct. The break-point decomposition, by contrast, records 4/5 milestones achieved and localizes the defect to "difference" interpretation, a diagnosable and distinct failure mode from early entity resolution.

Case 4. The four archetypes on a single task. This task, drawn from the Industry Planning category at medium difficulty, asks which of two strategic routes scores higher for the province where the food-and-beverage enterprise with the most cumulative Chinese invention patent grants is located. Across the ten evaluated models, six reach the correct answer of industrial chain extension route and four fail. Request counts span 4 to 83. We report five trajectories that cover all four archetypes from §4.1.

Case 4 (single task, five models compared)

Question. In 2022, in the food and beverage industry, for the province where the enterprise with the most cumulative Chinese invention patent grants is located, if that province chooses the "brand upgrade route" (market-cap-to-revenue ratio, profit margin, per capita market cap; weights 35%, 35%, 30%) versus the "industrial chain extension route" (total enterprises, revenue scale, upstream-downstream enterprise diversity; weights 40%, 30%, 30%), which route scores higher?

Reference steps.

1. From `company_operation_status.csv`, identify the food-and-beverage enterprise with the most cumulative Chinese invention patent grants = Qingqing Jinyin Food Company, 644 patents.
2. From `company_profile.csv`, obtain the company's province = Beijing.
3. From the regional aggregates and `company_profile.csv`, compute Beijing's six per-route metrics across market-cap-to-revenue ratio, profit margin, per capita market cap, total enterprises, revenue scale, and upstream-downstream diversity, then apply the two weighted formulas after cross-province min-max normalization.
4. Brand upgrade route score = 25.0, industrial chain extension route score = 83.1.

Gold answer. `Industrial chain extension route`.

Claude Opus 4.6 (20 requests, correct), the decisive solver.

1. Lists the database directory, then probes four CSV schemas in parallel and isolates `Y_EC_44` as the cumulative-patent target field and `industryId=10` as the food-and-beverage industry.
2. Sorts the matching companies by patent count and lands on Qingqing Jinyin Food Company at 644 patents on the first sort, reads off the company's province as Beijing.
3. Pulls Beijing's six per-route metrics from the regional aggregates, normalizes across the provinces with complete data, and computes both route scores.
4. Returns the industrial chain extension route after one self-consistency check on the metric definitions.

Minimax M2.7 (42 requests, correct), persistent-but-late.

1. Spends 41 silent assistant turns and roughly 60 tool calls scanning every profile and operation file for combinations of food, beverage, patent, and per-province aggregates before producing any user-facing text.
2. Surfaces Qingqing Jinyin Food Company in Beijing during the silent scan.
3. Computes the route scores directly from raw aggregate values, then re-checks ownership-type counts to reconstruct the upstream-downstream diversity metric.
4. At roughly twice Claude's request count, emits a single consolidated answer at the final turn that nominates the industrial chain extension route.

DeepSeek-V3.2 (83 requests, incorrect), wasteful trial-and-error.

1. Locks onto Yili Weiwei Wine Company in Hubei at 324 patents as the patent leader, missing the higher-patent Qingqing Jinyin entry under the food sub-industry. M1 already broken.
2. Burns most of its remaining request budget trying to reconstruct Hubei's per-route metrics from company-level data with mismatched units, then switches to regional aggregates.
3. Cannot align targetName variants across provinces and concedes "No relevant data found" after burning the largest request budget on the task.

Qwen3.5-Plus (56 requests, incorrect), wasteful trial-and-error.

1. Misreads the question's granularity, aggregating patent counts at the province level instead of selecting the individual top-patent enterprise.
2. Identifies Shanghai as the province with the highest aggregate patent count and attempts to compute the two routes for Shanghai.
3. Cannot recover an upstream-downstream diversity field, falls back to "No relevant data found" rather than reformulating the entity-selection step.

Kimi-K2.5 (4 requests, incorrect), disengaged.

1. Lists the database directory and issues five schema-probe tool calls across the next two assistant turns, all silent and without committing to a candidate enterprise.
2. Returns a fourth assistant turn that emits neither tool calls nor text, leaving every gold milestone unachieved.

Style-contrast analysis. The five trajectories trace the four archetypes from §4.1 on a single concrete task. Claude commits to one entity-selection hypothesis on its first sort and rides it to the answer in 20 requests. Minimax doubles the request count by exhaustively pre-scanning before committing and arrives at the same answer. DeepSeek and Qwen each burn fifty-plus requests on a wrong entity-selection branch, with DeepSeek picking the wrong individual enterprise (Hubei rather than Beijing) and Qwen picking the wrong abstraction level (provincial aggregate rather than top enterprise). Kimi terminates after four requests without ever attempting analysis. The correct-versus-incorrect split on this task is an entity-selection gap rather than a reasoning-depth gap. Every model that locked onto the right enterprise on its first or second attempt produced a correct answer, and every model that committed to a wrong entity propagated the error through dozens of request cycles or skipped the work entirely.

G Limitations and Future work

DataClawBench currently focuses on structured and unstructured textual modalities. Extending to richer modalities would better reflect real-world analytical workflows. Additionally, the milestone annotations are produced through human-in-the-loop pipelines, meaning that task decomposition and labeling granularity are inevitably subject to annotator judgment. The resulting milestones should therefore be treated as one plausible reference rather than a unique ground truth. Finally, A

limitation of our current process evaluation is that GPR and TPE are computed only on tasks with incorrect final answers. As a result, these metrics do not fully capture cases where an agent arrives at the correct answer through incomplete, accidental, or flawed reasoning. Evaluating such cases would require more fine-grained and rigorous process-level assessment, which we leave for future work.