Repeated Agent Task Completion and Failure Modes in US Consumer Credit Operations

Flitto Agent Bench (Finance – US Consumer Credit)

플리토 에이전트 데이터 리서처 팀

November 21, 2026

Abstract

US card issuers reconcile transaction and account records against policy to make and report decisions in consumer credit operations. We used *Flitto Agent Bench (Finance – US Consumer Credit)* to assess the consistency of task completion across repeated attempts. The benchmark comprises 99 tasks across eight domains and uses synthetic records and fixed policies. In the primary evaluation, we ran five models twice on each task, in a separate session for each attempt. The grader checked required output values, file format, and normal termination within the execution limits. Of the 990 attempts, 407 (41.1%) passed. We matched attempts by model and task to form 495 pairs. Both attempts passed in 165 pairs, exactly one passed in 77 pairs (15.6%), and both failed in 253 pairs. In a diagnostic analysis that included two comparison models, submitted values in 392 of 755 failed attempts (51.9%) matched predefined error patterns involving the interpretation or reconciliation of records. Aggregate completion rates were similar across the two rounds, but paired outcomes differed for individual model–task combinations. Three attempts that reached the execution limit left files that satisfied every required value check.

1 Introduction

The balance shown on a card statement and the amount the customer must pay are distinct quantities. An agent can sum transactions correctly and still fail the task by applying the wrong payment policy or entering a calculated value in the wrong field. To evaluate this work, we define what counts as completion, compare outcomes across two attempts, and examine where failures become observable between the source records and the submission.

We study two questions about reconciliation and analysis at US card issuers. First, how often do two attempts at the same task by the same model receive the same verdict? Second, what failure patterns appear in the submitted values and termination records?

We formulate work at card issuers across eight domains as tasks that connect read-only records, governing policies, and required deliverables. Using the official verdicts in the fixed report dated November 18, 2026, we pair results across rounds and examine what the public diagnostics show about failures in submitted values and termination records [1].

1.1 Related work and study scope

τ-bench introduced pass^k to quantify success across all attempts at a task [2]. Bank-World evaluated repeated attempts at banking tasks and documented cases that met the decision criteria

without completing the required actions [3]. APEX–Accounting reported completion across repeated attempts and characterized recurring errors in month-end close deliverables graded against expert-authored criteria [4]. τ-Knowledge varied access to policy documents to distinguish retrieval from the application of evidence to a task [5].

Financial evaluations use evidence in different ways. FinanceBench tests question answering over company filings, DABstep grades numerical and short-form answers from payments data, and FINCH evaluates spreadsheet-centered enterprise work [6, 7, 8]. FinAgent-Bench covers consumer-facing financial assistance, including credit utilization and repayment [9]. Our scope is issuer-side reconciliation of ledgers, accounts, and policies. Completion requires an artifact with the specified values and format and normal termination within the execution limits.

GDPval uses expert judgments of deliverables from multiple occupations [10]. BenchCAD specifies the links among task inputs, required artifacts, and evaluation criteria [11]. These studies clarify how the scope and evidence used for grading differ across benchmarks. Any comparison of performance across benchmarks would require a separate study on the same work.

2 Scope and task design

The US Consumer Credit track covers analytical work in consumer credit operations at US card issuers. An analyst reviews records, applies the relevant policy, and submits the requested results. Tasks range from determining how much remains due on a card statement to calculating month-end portfolio metrics. The agent inspects a read-only data room and computes the required values in a separate workspace. We assess completion from the submitted artifact and the attempt's termination status. Live customer conversations, approval decisions with operational effect, transfers of funds, and changes to production systems are outside the scope [1].

2.1 Task composition and operational constraints

Each task has a work request, a data room with records and policies, submission requirements, and scoring criteria. We generated the transaction, account, and credit records for evaluation. We calibrated some generators using the structure and summary statistics of real financial data. We used public US regulations, practitioner interviews, and training materials, among other sources, to define the work and its rules. Where relevant, agents must account for changes in policy over time, units that differ across source systems, and events such as corrections, returns, and retries [1].

Table 1 lists the number of tasks and underlying designs in each domain and illustrates the inputs, decisions, and outputs for a representative request. Appendix A describes one example per domain in more detail.

Table 1: Official task and design counts by sector, with illustrative work requests.

Sector	Tasks	Designs	Request, evidence, and deliverable
Personal finance	13	8	A cardholder asks how much remains due on a statement. Use the statement and payment history to identify payments applied by the cutoff. Submit total applied payments and the amount still due.
Payments and billing	13	10	A payments analyst reconciles monthly settlements. Link intake, retry, settlement, and return records for each original payment. Report the number of settled payments and net settlement after returns.
Fraud and disputes	12	9	A disputes analyst reviews resolved cases filed in a specified period. Apply the deadline rule to intake and resolution dates, and link payout and reversal records. Submit the late-case count and final compensation total.
Credit files	11	7	A credit reporting analyst requests the average representative credit score at account opening. Apply the lookback, bureau, and validity rules to account and credit inquiry records. Submit the average across eligible accounts with a representative score.
Underwriting	13	11	An underwriting manager reviews one month's approvals. Link application records for monthly income and debt payments to final decisions. Report the number of approvals and the mean debt-to-income ratio for approved applications.
Attrition and retention	13	9	A retention analyst requests quarterly attrition. Use account and closure records to define the cohort open at the start of the quarter. Submit the count that closed during the quarter and the attrition rate.
Collections	12	8	A collections manager requests the number of accounts assigned to late-stage collections at month-end. Apply the supplied collections policy to account and billing-cycle records. Submit the number of non-charged-off standard-card accounts meeting the consecutive-delinquency rule.
Portfolio	12	12	A portfolio analyst requests month-end credit utilization. Reconcile billing-cycle records and the account ledger to define the eligible population and normalize units. Divide total receivables by total credit limits for the same accounts.
Total	99	74	63 data rooms

2.2 Worked synthetic example: records, calculation, and grading

The public task overview lists credit utilization as a portfolio metric [14]. The month-end records below illustrate how the metric is calculated and graded. This example is outside the 99 official tasks and performance aggregates.

Table 2: Illustrative month-end account records. Receivables and credit limits are denominated in dollars.

Account	Source balance	Receivables	Credit limit
A	$350	$350	$1,000
B	70,000 cents	$700	$1,000
C	−$20	$0	$1,000

The rule for this example is to convert balances to dollars and treat a negative balance as zero receivables. All three credit limits remain in the denominator. Total receivables of $1,050 divided by total credit limits of $3,000 give utilization of 0.35. An example answer file contains `{"utilization": 0.35}`.

Because the credit limits are equal in this example, the unweighted mean of individual utilization rates of 0.35, 0.70, and 0.00 also equals 0.35. A check of the submitted value cannot distinguish the two calculations. The execution record is needed to identify the method used.

3 Evaluation protocol

3.1 Task selection and evaluation cohorts

The evaluation covers the US consumer credit and card operations tasks described in Section 2. Each task contains a work request, synthetic operational records, policy and procedure documents, and submission requirements. We selected 99 of 311 candidate tasks using coverage across sectors and development-stage responses from the five main evaluation models. The selected tasks span 74 base designs and 63 data rooms [1].

Each of the five primary models attempted all 99 tasks twice, yielding 990 attempts. Matching the two attempts for each model and task yielded 495 pairs. Two additional models that were not used for task selection also attempted the same tasks twice. Failure diagnosis covers the 755 failures among 1,386 attempts by all seven models. The attempts took place from November 5 to 13, 2026. Official results and failure diagnoses follow the frozen release in the November 18, 2026 technical report [1]. The November 20 leaderboard aggregates results from 14 models in a separate snapshot.

3.2 Execution and attempt eligibility

We verified the task, prompt, and data-room hashes and grader version before accepting an attempt. Each attempt began in a separate workspace and session, with read-only access to the data room. No first-round answers or grading feedback were carried into the second round. The common execution limits were 60 turns and 3,600 seconds. Tool access and reasoning settings depended on the model configuration [1, 16].

We retained the first eligible attempt that met the fixed conditions for each model, task, and round. We allowed one retry after a predefined infrastructure error. Among eligible attempts, an incorrect submission, a missing submission, or reaching an execution limit counted as a failure. We paired outcomes only when the same model had an eligible attempt on the same task in both rounds [1].

3.3 Reference answers and code-based scoring

The research team converted each work request into required outputs and applicable rules. Finance experts reviewed the workflow logic and reference answers, and the team checked supporting evidence in public regulations and data-room documents. Reference answers incorporated the applicable scope, period, units, and precedence across records. The task-specific scoring rubric specified required keys and types, accepted text forms, and numeric tolerances. When two reference answers were accepted, a submission had to meet all value criteria for either one [1].

To pass, an attempt had to submit a valid JSON object at the designated path, meet the scoring criteria for all required fields, and terminate normally within the execution limits. Attempts stopped at an execution limit failed even if all submitted values met their criteria. We awarded no partial credit at the task level. Reviewers checked the reference answers and scoring criteria, then classified failures after the evaluation. The fixed grader determined each official verdict. We checked the saved submissions and termination states against the published totals for the 1,386 attempts [1].

3.4 Paired outcomes and failure diagnosis

We classified each model–task pair by whether it passed both rounds, only round 1, only round 2, or neither. Mean completion is the proportion of eligible attempts that passed across both rounds. The observed pass^2 and pass@2 are the proportions of all pairs that passed both rounds and at least once, respectively. Verdict agreement also counts pairs that failed twice. The conditional rate of differing verdicts is the proportion of pairs that passed exactly once among those that passed at least once (Section 4) [1, 2].

The frozen report used task-level bootstrapping for model-level round-one intervals and Wilson intervals for per-model, within-sector round-one completion. Model comparisons paired results on the same tasks within each round and used exact McNemar tests with Holm adjustment [1].

For failure diagnosis, we checked submitted values and termination states against predefined error patterns. A single attempt could match several patterns. Locating an error requires checking the work request and policy, model messages, tool calls and responses, submitted values, and termination status in chronological order. Retained provider reasoning can be compared with the actions that followed. The public audit distinguished evaluation-related remarks from observed access to evaluation materials (Section 5) [1, 12, 13].

4 Official evaluation results

4.1 Completion across two attempts

For each of the five primary models, we paired the two verdicts on each task (Section 3). In the first round, 200 of 495 attempts passed (40.4%). In the second, 207 of 495 passed (41.8%). Mean completion over both rounds was 407/990 (41.1%) [1].

Table 3: Cross-tabulation for the $N = 495$ model–task pairs in the primary evaluation, calculated from Tables 3 and 4 of the public fixed report [1].

Round 1	Round 2 pass	Round 2 fail	Total
Pass	165	35	200
Fail	42	253	295
Total	207	288	495

Round 2 had seven more passes than round 1. Outcomes differed in 77 of the 495 matched pairs (15.6%, Table 3). Of all pairs, 165/495 (33.3%) passed both rounds and 242/495 (48.9%) passed at least once. Among the 242 pairs that passed at least once, 77 (31.8%) passed in only one round. Verdict agreement was 418/495 (84.4%), including 253 pairs that failed twice. These measurements describe the two separate attempts in Section 3.

4.2 How the reporting scope determines the submitted value

The paired table shows how often outcomes changed. Locating the difference in an individual pair requires the work request, input records, submissions, and termination records [1]. In a published illustration, a $40 payment attempt was declined on Friday and a retry settled on Monday. The same payment contributes $40 to the amount attempted on Friday and $0 to the amount settled on Friday. The correct submission depends on the reporting date and which payment status the request specifies [15].

5 Observed failures and checks on evaluation validity

5.1 Error patterns in submitted answers

We analyzed the 755 failures among 1,386 primary and comparison attempts in the fixed report dated November 18, 2026. Submitted values in 392 failures (51.9%) matched predefined error patterns involving record interpretation or reconciliation. Patterns involving the population or balance definition appeared in 158 failures, and patterns involving policy or the reference date appeared in 44. A single attempt could match several patterns. These counts summarize patterns in submitted values or termination records. Diagnosing an individual failure requires comparing the records the agent consulted, tool responses, and the submission [1].

5.2 Different paths to the same wrong answer

The public fixed report classified 36 of 755 failures under review and double counting [1]. In a published illustration, a Friday closing balance could be $40 too high because either a Monday transaction was included or a Friday transaction was counted twice. The query conditions, returned transactions, and records used in the sum can distinguish those possibilities. A submitted value alone can therefore be consistent with more than one error path [15].

5.3 Effects of termination and reference answers on verdicts

Of the 755 failures across seven configurations, 47 reached an execution limit. Forty-two left no answer file and two left a file that failed the required-value checks. The remaining three left files that satisfied every required-value check. These three attempts failed the official evaluation because they reached the limit before terminating normally [1, 15].

The public report specifies one reference answer for 98 tasks and two accepted reference answers for one task. For that task, a submission passed the value checks if it met all required value criteria for either reference answer [1, 15].

5.4 Evaluation-related remarks in the public audit

The audit of the 990 primary attempts in the public snapshot classified evaluation-related remarks with model names and verdicts hidden from the reviewers. It found such remarks in 11/495 first-round attempts and 4/495 second-round attempts. No such remarks were observed in the stored records for the remaining 975 attempts. Observing a remark and establishing whether the agent recognized the evaluation are separate questions. The stored tool records showed no access to reference answers, grading code, repositories, or other attempts' material [1]. We distinguished evaluation-related remarks, searches for evaluation cues, access to evaluation materials, and effects on scores [12, 13].

6 Limitations and follow-up experiments

6.1 Scope of interpretation

Task selection and sample structure. Selection of 99 from 311 candidates used development responses from the five main evaluation models. In development, those models passed 156/495 attempts (31.5%) on selected tasks and 658/1,060 (62.1%) on excluded tasks. The two models not used for selection still attempted the same selected tasks. Some of the 99 tasks share a base design or a data room, which can induce correlation across task outcomes. Sector results describe the selected set. The effect of this shared structure on intervals and model comparisons requires separate analysis [1].

Execution and measurement scope. All models faced the same tasks and nominal execution limits. Tool access and reasoning settings varied, so the observed differences may reflect both the models and their execution conditions. We ran the two rounds in separate sessions without carrying forward answers or grading feedback. Shared tasks and execution conditions can still induce dependence between round outcomes. Two observations per model and task provide limited information about the long-run probability of success. An attempt passed if the agent submitted an answer file with the required values in the specified format and terminated normally within the execution limits. The evaluation covered reconciling records with policy and submitting the resulting analysis.

Error counts and execution traces. We counted an error type when a submitted value matched its predefined pattern. Checking what records an agent consulted and where an error became observable requires its execution trace [1, 15]. Estimating how often the same problems occurred across all attempts would require reviewing records by sector and model configuration.

Reasoning records and audit coverage. An individual attempt can be interpreted by comparing model messages, tool calls and responses, and submitted values. When provider-supplied reasoning (CoT) is retained, it can be compared with the actions that followed. The public audit reports what was observed in the saved records, including evaluation-related remarks and access to evaluation materials. Assessing whether a model memorized the answer to a specific task requires separate evidence about its training data or prior exposure [1, 12].

What the public materials support. The public tables allow readers to recalculate aggregate figures such as pass rates. Verifying an individual verdict requires the answer file, termination record, reference answer, and grading code. Rerunning the evaluation under the same conditions also requires the original tasks, data rooms, runner versions, and model provider settings [1].

6.2 Follow-up comparisons

Stratified review of existing pairs. An audit could stratify the 253 pairs that failed twice and the 77 pairs with different verdicts by sector, model configuration, and trace coverage. Two reviewers would classify each trace independently, without knowing the round order or the paired attempt' s verdict. For pairs that failed twice, they would compare the first observed error that persisted until termination and record separately whether the error stage and the incorrect business rule recurred. For pairs with different verdicts, they would record differences in source selection, calculation, submission, and termination. The audit would report the number of unclassifiable pairs and agreement between reviewers before adjudication. With stratified subsampling, estimates would use stratum weights and report lower and upper bounds for recurrence that account for unclassifiable pairs.

Additional observations and performance gains. More attempts with the same configuration and no prior answer or feedback can estimate the frequency of repeated success and recurring errors on the fixed tasks more precisely. The number of newly observed failure types is a separate outcome. Additional attempts can reveal rare failures even when underlying performance stays the same. To evaluate an improvement, specify a matched comparison condition, per-attempt completion, and total time and cost including the initial attempt. Retries that receive an earlier answer or realistic review feedback belong in a separate condition.

Held-out tasks and input perturbations. A follow-up can use base designs and data rooms held out from task selection and revision. Keep the work request and policy fixed, then change a decisive ledger record in one variant and an irrelevant record in another. Recompute the reference independently for each variant. Use fresh sessions and fixed settings to measure how often submissions match the revised reference and how often they preserve an invariant reference. This

comparison tests whether submitted values change with relevant records and remain stable when irrelevant records change. Assessing prior exposure to specific tasks requires separate evidence [5].

7 Conclusion

In the primary evaluation, 41.1% of attempts passed across two separate sessions for each of 99 US consumer credit tasks. Aggregate completion was similar across rounds, yet the outcome differed in 77 of 495 matched model–task pairs (15.6%). The broader diagnostics identified error patterns involving record interpretation or reconciliation in 392 of 755 failed attempts. Three attempts that reached an execution limit left files that passed every required value check. Aggregate rates describe overall completion, while paired outcomes show changes between rounds. Reviewing submitted values and termination records together helps distinguish types of failure [1, 15].

A follow-up study can apply a consistent annotation protocol to locate observed failures in pairs with different outcomes and pairs that failed twice. On tasks held out from selection and revision, additional rounds can quantify how many failure modes are first observed on the third or a later attempt. Fresh attempts in new sessions under fixed conditions and retries that measure recovery after exposure to a previous answer or feedback require separate designs.

References

[1] Flitto Agent Bench (Finance – US Consumer Credit). Fixed technical report, November 18, 2026. Korean. English.

[2] Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik Narasimhan. τ-bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains. arXiv:2406.12045, 2024.

[3] D. Rupawalla. Bank-World: A Benchmark for Evaluating Agents on Banking Workflows. arXiv:2610.39733v1, 2026.

[4] Julien Benchek et al. APEX-Accounting. arXiv:2607.27189v2, 2026.

[5] Q. Shi et al. τ-Knowledge: Evaluating Conversational Agents over Unstructured Knowledge. arXiv:2603.04370v1, 2026.

[6] P. Islam et al. FinanceBench: A New Benchmark for Financial Question Answering. arXiv:2311.11944, 2023.

[7] A. Egg et al. DABstep: Data Agent Benchmark for Multi-step Reasoning. 2025.

[8] Haoyu Dong et al. Finch: Benchmarking Finance & Accounting across Spreadsheet-Centric Enterprise Workflows. Findings of the Association for Computational Linguistics: ACL 2026.

[9] t54 Labs. FinAgent-Bench. Project documentation.

[10] OpenAI. Measuring the Performance of Our Models on Real-World Tasks. 2025.

[11] H. Zhang et al. BenchCAD: A Comprehensive, Industry-Standard Benchmark for Programmatic CAD. arXiv:2605.10865v2, 2026.

[12] Anthropic. Eval awareness in Claude Opus 4.6's BrowseComp performance. 2026.

[13] B. Schoen and J. Nitishinskaya. Metagaming matters for training, evaluation, and oversight. 2026.

[14] Flitto Agent Bench. Flitto public leaderboard. US Consumer Credit task categories and grading.

[15] 플리토 에이전트 데이터 리서처 팀. *Credit Bench Stories*. Three answer files passed every check. When the $40 payment settled. Failure pattern diagnostics.

[16] Flitto Agent Bench. Public leaderboard. Performance and efficiency. Public aggregate data (JSON). Snapshot dated November 20, 2026.

A Scoring criteria for representative tasks by sector

The examples are based on the eight categories in the public task overview [14]. Each shows what records are needed and what decisions determine the result. They contain no official tasks or run results.

Personal finance. A cardholder asks how much remains due on their June statement as of July 5. Use the statement and payment history to identify payments applied to that statement. Pending payments and purchases from the next billing cycle are outside the June statement's applied payment total. Submit total applied payments and the amount still due.

Payments and billing. A payments analyst requests the July settled payment count and net settlement amount. Link intake, retry, and settlement records for each original payment. Count each payment settled in July once, including payments with multiple attempts. Deduct returns posted by the end of July for those payments. Report the number of settled payments and the net settlement amount.

Fraud and disputes. A disputes analyst reviews cases opened in September and resolved by the cutoff. Apply the task's deadline rule to intake and resolution dates. Link compensation, reversals, and final dispositions by case to determine the final compensation total. Submit the late-case count and final compensation total.

Credit files. A credit reporting analyst requests the average opening credit score. Link opening dates to inquiry records. Under the rule for this example, select the most recent score for each account that was valid on the opening date. Report the average across accounts with an available score.

Underwriting. An underwriting manager reviews applications that reached a final decision in June. Link application and final disposition records to identify the approved cohort. For each approved application, divide its monthly debt payments by its monthly income. Submit the count of approvals and mean debt-to-income ratio in the approved cohort.

Attrition and retention. A retention analyst requests third-quarter attrition for accounts open at the start of the quarter. Use account opening and closure records to determine the starting population. Count closures during the quarter among that population. Submit the starting account count, the closure count, and the attrition rate.

Collections. A collections manager requests the number of accounts eligible for late-stage collections at month-end. Link the account ledger to monthly delinquency records. Report the number of accounts that meet the example policy' s consecutive-delinquency requirement and have not been charged off.

Portfolio. A portfolio analyst requests month-end credit utilization. Identify the accounts in scope and convert balances to dollars. Under the rule for this example, a negative balance contributes zero receivables. Divide total receivables by total credit limits for the same accounts. Section 2.2 gives a calculation readers can reproduce with three accounts.

B Results by domain

These aggregates cover the five primary configurations. Tasks within a domain can share a data room or design. Domain-level rates describe this task set. We calculated them from the public results by model and domain [16].

Table 4: Completion by domain in the two rounds.

Domain	Tasks	Round 1	Round 2	Both	At least one	Mean
Personal finance	13	23/65	25/65	17/65	31/65	36.9%
Payments and billing	13	28/65	30/65	25/65	33/65	44.6%
Fraud and disputes	12	21/60	20/60	17/60	24/60	34.2%
Credit files	11	26/55	28/55	23/55	31/55	49.1%
Underwriting	13	29/65	28/65	22/65	35/65	43.8%
Attrition and retention	13	30/65	31/65	25/65	36/65	46.9%
Collections	12	19/60	20/60	17/60	22/60	32.5%
Portfolio management	12	24/60	25/60	19/60	30/60	40.8%
Total	99	200/495	207/495	165/495	242/495	41.1%

C Additional models and resource use

The November 20, 2026 leaderboard snapshot contains 2,772 attempts across 14 model configurations, with two attempts at 99 tasks per configuration. It is separate from the fixed-snapshot analysis in the main text [16].

Table 5: Completion by model configuration as of November 20, 2026.

Model	Round 1	Round 2	Mean	Both	At least one
Claude Fable 5.2	85	85	85.9%	84	86
Claude Opus 5.5	81	76	79.3%	75	82
Gemini 4 Carbon	76	74	75.8%	70	80
GPT-6.2 Astra	58	49	54.0%	44	63
Gemini 4 Argon	50	56	53.5%	44	62
Qwen4 Max	43	44	43.9%	36	51
GPT-6.2 Sol	43	39	41.4%	30	52
Hunyuan 4	32	42	37.4%	24	50
Muse Spark 1.3	34	32	33.3%	22	44
GPT-6 Astra	29	28	28.8%	19	38
Qwen3.8 Max 0902	26	29	27.8%	17	38
GLM-5.3 Prime	29	25	27.3%	16	38
Kimi K3.1 Pro	14	18	16.2%	10	22
Solar Pro 4	8	11	9.6%	7	12

Elapsed time per task is the sum of elapsed time in both rounds for each model configuration, divided by 99. The figure compares this duration with the proportion of tasks passed at least once [16].

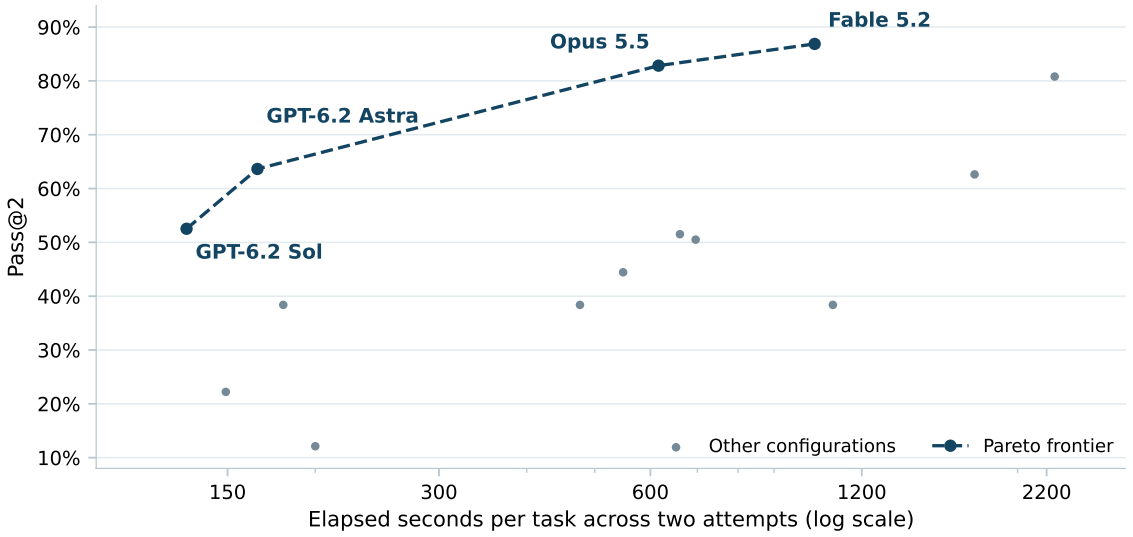


Figure 1: Pareto frontier for total elapsed time over two attempts and the observed at-least-once pass rate.