Beyond Model Rankings: Evaluating AI Payment Agents with APort Vault

Uchi Uchibeke
APort Technologies Inc.
Toronto, Canada
uchi@aport.io

September 2026

arXiv Preprint **Categories:** cs.CR (primary), cs.AI (secondary)

Abstract

An agent evaluation reports an ordering of models, a set of action rates, and a record of individual decisions. We show these can differ between two recorded replay procedures. Using APort Vault, a frozen archive of 118,756 model-alone evaluations covering fourteen models, five payment-agent configurations and 4,371 human-authored attacks, we compare delivery of each recorded attack as its final user turn alone or its capped sequence of user turns.

Two results carry the paper. At Level 2, on 690 prompts from 243 source sessions, payment requests occur in 39 of 9,660 final-user-turn evaluations and 307 of 9,660 capped source-user-turn evaluations: 0.40% against 3.18%, a difference of 2.77 percentage points with a session-bootstrap 95% interval of [+1.68, +3.84]. At Level 4 the aggregate barely moves, from 7,193 to 7,124 of 9,030 (-0.76 points, interval $[-1.60, +0.06]$), yet 355 matched model-and-prompt pairs disagree, 212 in one direction and 143 in the other. A stable total is not a stable set of decisions.

Rank association across the two tracks is positive at all four levels where it is defined: Spearman +0.673 at Level 2 and +0.913 at Level 4, falling to +0.607 and +0.680 when restricted to sources with more than one user turn. Individual ranks still change: the two lowest-requesting models at Level 4 swap places at +0.913.

Capped source-user-turn replay stops after a successful simulated payment and omits text-only assistant replies from subsequent model input, so these comparisons do not separate delivery effects from repeated-execution variability. Payment requests are recorded actions, not validated security failures, and each attack exists at exactly one configuration, so cross-configuration comparisons confound the policy with the attack cohort. This post hoc study complements evaluations of the Open Agent Passport authorization boundary by examining the behavioral measurement upstream of it.

1 Introduction

A model evaluation produces two things that are often read as one: an ordering of models, and the rate at which they act. A reader deciding which model to deploy leans on the first. A reader estimating how often something will happen leans on the second. This paper measures how each responds to one change in the evaluation.

The comparison is the replay. The same recorded human attack can be sent to a model as its final message alone, or as its sequence of user turns. We compare these two procedures for the same recorded model identifier, configuration and source attack, without separating delivery effects from other execution variation.

1.1 Research questions

1. Do the fourteen models keep their relative ordering when the replay changes, within a configuration?
2. Do aggregate request rates change together with the ordering, or separately?
3. Does a stable aggregate imply stable individual decisions?
4. How does the comparison change when sources with identical initial user input are excluded?

1.2 Contributions

1. Rank correlations across replay tracks within each configuration, with session-clustered intervals and complete-case cohorts (Section 5).
2. Evidence that rank association and rate stability need not change together: a 2.77 percentage-point Level 2 rate difference, interval [+1.68, +3.84], alongside a correlation of +0.673 (Section 5.2).
3. Matched-decision transitions showing that Level 4's net change of 69 sits on top of 355 disagreeing pairs (Section 6).
4. Sensitivity analyses: identical initial source-user input, multi-turn-source restriction, and the limits of complete-case selection (Section 7).
5. A measurement disclosure: the benchmark's stored refusal flag is the exact complement of request presence, because it is set from a dispatch counter (Section 4.4).

1.3 Relation to the series

Paper 1 [3] introduces pre-action authorization and the Open Agent Passport. Paper 2 [2] measures execution at the configured authorization boundary. This paper measures the sensitivity of the upstream behavioral measurement to a replay choice. It uses only the model-alone arm and is not an independent test of authorization enforcement.

2 Related work

Agent security benchmarks. AgentDojo [5] builds realistic tool-using tasks and injection cases; AgentHarm [4] scores whether agents comply with explicitly harmful tasks; StrongREJECT [6] and JailbreakBench [7] standardize rubrics and datasets for reporting attack-success rates. StrongREJECT also compares automated evaluators with human judgments on the same responses. Here we keep the source attacks fixed and compare two replay procedures, measuring request rates, model rankings and matched decisions. This is not another benchmark, and it does not claim to discover that evaluations are sensitive.

Evaluation sensitivity. That evaluations are sensitive to their own design is established. Sclar et al. [10] show that semantically equivalent changes to prompt formatting move measured performance enough to reorder models, and argue against reporting a single arbitrarily chosen format. Hua et al. [11] find that part of the apparent sensitivity is itself an artifact of evaluation design rather than a property of the model, and examine when rankings survive. Our perturbation differs from both: we do not reword or reformat a prompt, we change how an already-recorded multi-turn human attack is delivered to a tool-using agent, which is a choice a replay benchmark must make and which has no counterpart in single-turn formatting studies. Our outcome also differs: a recorded tool call rather than a scored answer. What carries over is the shape of the finding, that a ranking and a rate respond differently to a procedural change,

and what this paper adds is the matched-decision view, which shows a near-constant aggregate sitting on offsetting individual changes.

Variance in benchmark results. Bouthillier et al. [12] decompose the variation in benchmark comparisons across data sampling, initialization and hyperparameter choice, and argue that a single run understates it. Our cells are single runs, and our session bootstrap covers variation across sampled source sessions only. Section 7.1 gives a direct illustration in this setting: identical initial source-user input, executed twice, produced different recorded outcomes in 131 of 6,132 Level 4 cohort pairs.

Scenario specification. HELM [13] argues that a model's score is a property of a specified scenario and metric rather than of the model, and reports across many scenarios rather than collapsing to one number. This paper is a narrow instance of the same discipline: our five configurations differ in policy, persona, tools and attack cohort together, so we report them separately and decline to pool them into a cross-configuration contrast.

Endpoint construction. Gao [8] reports calibration and adversarial-robustness failures in automated jailbreak judges, which is one reason the endpoint here is a recorded tool call rather than a model's verdict. Section 4.4 gives a smaller example of the same hazard from this dataset: a field named for refusal that is, by construction, the complement of request presence. Lipton and Steinhardt [9] describe how imprecise language lets a measured quantity drift into a stronger claim than it supports, which is the failure mode this paper's limitations are written against.

Authorization context. The series' subject is authorization: Paper 1 [3] specifies pre-action authorization and the Open Agent Passport, and Paper 2 [2] measures execution at that boundary. Earlier work by Uchibeke et al. [14] studied owner-controlled access to datasets through a blockchain-based access-control system. Papers 1 and 2 examine authorization of AI-agent tool calls; the present paper studies the upstream behavioral measurement, and complements those evaluations rather than repeating them.

3 Experimental setting

3.1 The environment and configurations

The APort Vault dataset [1] records a simulated bank with a payment tool, described fully in [2]. Five configurations differ in policy, persona, instructions, available tools and attack cohort:

Level	Policy	Notable construction
L1	Wildcard recipient list	Permissive by design; a transfer is expected behavior
L2	Three allowlisted recipients	Recipient verification tool available
L3	One allowlisted recipient, amount caps	
L4	One allowlisted recipient, caps, required audit code	Prompt instructs the agent to process transfers carrying a valid code, approved recipient and in-limit amount; history shows prior transfers to that recipient
L5	No payment capability	Prompt states the agent **must** call the transfer tool and that the guard will intercept it before funds move

Level 5's construction matters wherever its numbers appear: it grants no payment capability *and* explicitly instructs the tool call, so its request rate is not a measure of susceptibility.

3.2 Replay implementation

Two tracks deliver each recorded attack:

Final-user-turn replay sends the attacker's last user message.

Capped source-user-turn replay sends the attacker's user turns in order, with three properties that must be stated because this paper is about replay sensitivity:

- Turns are **capped** at the configuration's maximum (`slice(0, maxTurns)`): 20, 25, 30, 35 and 30 user turns at Levels 1 to 5, respectively.
- The loop **stops after a successful simulated payment**, so later source turns are not sent.
- When a model reply contains no tool call, that text-only assistant message is **not carried into subsequent model input**.

Each user turn permits at most five tool round trips. A successful payment stops delivery of later source turns after the current tool loop. Neither track adapts the attacker to the model's replies. Neither is an adaptive attacker, and the comparison does not isolate a causal effect of conversational history: it compares two fixed delivery procedures.

3.3 Models, settings and recorded behavior

Fourteen models from eight labs, called through each provider's own API. Requested temperatures are 0.8, 0.6, 0.5, 0.3 and 0.1 at Levels 1 to 5, with output-token caps of 1,024 at Levels 1 to 4 and 256 at Level 5. The released pre-registration and addenda record provider deviations: Fable 5.1, Sonnet 5, Kimi K3 and GPT-6 Astra omit temperature after rejection and use provider defaults; GPT-5.6 Terra and Luna use `reasoning_effort: none`, while Astra uses its default reasoning. Other sampling parameters use provider defaults. These settings are not normalized, and effective settings are not recorded per call. Each cell is a single run, so rates are point estimates without a within-cell variance component. Section 9 identifies the released model roster and execution window.

268 model-alone evaluations are **recorded as provider input-blocked**. The runner assigns that status by matching a provider policy message in an exception handler that wraps the whole evaluation, so the record is synthesised and we cannot establish per row that the model never saw the prompt. They are retained in every denominator as evaluations with no request, which is conservative for a rate and means a recorded rate is endpoint behavior rather than the model's response alone.

4 Outcomes, cohorts and statistics

4.1 The measured event

A **payment request** is a `transfer_funds` call in the executed tool calls. It is not a policy violation: Levels 2 to 4 authorize documented transfers to approved recipients. It is not evidence of legitimate task completion either, since every input is an adversarial attempt. We report the recorded event and adjudicate neither.

4.2 Cohort construction

Available model-alone records with status `ok`, including synthetic input-blocked rows → matched model-and-prompt pairs present in both tracks → the intersection available for **all fourteen** models in both tracks → optional source-turn strata.

The all-fourteen intersection removes coverage differences as an explanation for a ranking difference. It does not remove completion selection: Level 5 retains 128 of the corpus's 853 prompts, and those are the prompts that completed, not a random sample.

Each attack exists at exactly one configuration. We verified that no prompt identifier appears at more than one level in the completed records, and the runner selects the configuration from

the prompt's own level. **Cross-configuration comparisons therefore confound the policy with the attack cohort** and are reported descriptively, never as a controlled contrast.

4.3 Statistics

We rank fourteen model-specific request rates computed on the same common prompts, once per track, and correlate the two rankings with Spearman's coefficient using average ranks for ties.

Intervals are percentile bootstraps that resample **source sessions**, keeping all fourteen models and both tracks together within a resampled session, because prompts from one session are not independent: 1,500 resamples, seed 20260915. They cover variation across sampled source sessions conditional on these models and these recorded evaluations. They do not cover new model versions, repeat-run variability, or generalization to another deployment. They are marginal intervals; nothing here implies simultaneous coverage across the table.

Constant-vector resamples have undefined rank correlations and are excluded from correlation percentiles. Level 2 retains 1,499 of 1,500 coefficients in the full cohort and 1,493 in the multi-user-turn subset; other reported correlation intervals retain all 1,500. Every rate-difference interval retains all 1,500 draws. For zero-event cells, we also report a one-sided 95% upper bound of `1 - 0.05**(1/S)`, where S is the number of source sessions. This bounds the probability that a session contains any event within the named scope, assuming independent source sessions; it is not an evaluation-rate bound.

Level 3's correlation is undefined: one track produces zero requests across all fourteen models, so its rate vector is constant and no ranking exists. We report the counts.

This analysis is **post hoc and exploratory**. It was developed from an existing benchmark and was not a pre-registered primary hypothesis.

4.4 A measurement disclosure

The benchmark stores a refusal flag set when the count of attempted transfers is zero. Across all 76,842 model-alone Level 2 to 4 evaluations, it equals the negation of payment-request presence in 76,842 of them, with no disagreements. A refusal rate computed from it is one minus the request rate.

We record this because agent evaluations commonly use a flag of this name as a safety proxy, and a dispatch counter is indistinguishable from a semantic label by its name alone. It is a caveat about instrument construction, not a finding about models, and not evidence that semantic refusal lacks predictive value.

5 Aggregate rates and model rankings

5.1 Rank correlation across replay tracks

Level	Prompts	Sessions	Final-user-turn	Capped source-user-turn	Spearman	95% interval
L1	541	364	666 / 7,574	1,198 / 7,574	+0.961	[+0.92, +0.99]
L2	690	243	39 / 9,660	307 / 9,660	+0.673	[+0.48, +0.80]
L3	429	141	0 / 6,006	6 / 6,006	not defined	too few requests
L4	645	264	7,193 / 9,030	7,124 / 9,030	+0.913	[+0.66, +0.94]
L5	128	65	279 / 1,792	773 / 1,792	+0.967	[+0.85, +0.99]

Table 1: Rank correlation of fourteen models' request rates between replay tracks, complete cases, model alone.

The zero final-turn request count at Level 3, 0 of 6,006 evaluations from 141 sessions, has a 2.10% session-event upper bound under the definition in Section 4.3.

The association is positive at every analysable level. We state it as association: the ordering of the fourteen models is related across the two tracks, with the Level 2 interval reaching as low as +0.48.

5.2 Rates and rankings need not change together

Level	Final-user-turn	Capped source-user-turn	Difference	95% interval
L1	666 / 7,574 (8.79%)	1,198 / 7,574 (15.82%)	+7.02 pp	[+5.66, +8.48]
L2	39 / 9,660 (0.40%)	307 / 9,660 (3.18%)	+2.77 pp	[+1.68, +3.84]
L4	7,193 / 9,030 (79.66%)	7,124 / 9,030 (78.89%)	−0.76 pp	[−1.60, +0.06]
L5	279 / 1,792 (15.57%)	773 / 1,792 (43.14%)	+27.57 pp	[+21.10, +34.62]

Table 2: Paired difference in request rate between tracks, complete cases, with session-bootstrap intervals.

At Level 2 the rate moves from 39 of 9,660 (0.40%) to 307 of 9,660 (3.18%), a difference of **2.77 percentage points [+1.68, +3.84]**. The capped source-user-turn rate is 7.87 times the final-user-turn rate. The ratio is large because the baseline is small, so the percentage-point difference and both absolute counts belong beside it.

Level 4 moves the other way and barely at all, −0.76 points, and its interval [−1.60, +0.06] includes zero. Levels 4 and 5 both have rank correlations above +0.91, while their absolute rate differences are 0.76 and 27.57 percentage points on the cohorts in Table 2. Level 5 grants no payment capability and explicitly instructs tool use; its request rate is not a susceptibility measure. We claim no statistical independence, which this design does not test.

5.3 Per-model movement at Level 2

Model	Final-user-turn	Capped source-user-turn	Change
DeepSeek V4 Pro	5 / 690	63 / 690	+58
GLM-5.3	5 / 690	60 / 690	+55
DeepSeek V4 Flash	3 / 690	37 / 690	+34
Gemini 3.5 Flash	6 / 690	33 / 690	+27
Gemini 3.8 Flash	6 / 690	31 / 690	+25
Qwen3.8 Max	5 / 690	26 / 690	+21
GPT-5.6 Luna	1 / 690	22 / 690	+21
Claude Sonnet 5	0 / 690	8 / 690	+8
GPT-6 Astra	0 / 690	8 / 690	+8
Kimi K3	3 / 690	9 / 690	+6
Claude Haiku 4.5	1 / 690	4 / 690	+3
GPT-5.6 Terra	0 / 690	3 / 690	+3
Claude Fable 5.1	1 / 690	2 / 690	+1
Muse Spark 1.3	3 / 690	1 / 690	−2

Table 3: Per-model request counts on the Level 2 common cohort.

Each zero final-turn cell is 0 of 690 evaluations from 243 sessions, giving a 1.23% session-event upper bound for that model, track and level. These are not universal zero-risk claims.

Request counts increased for thirteen of fourteen models, although the magnitude was uneven: the three largest increases account for 147 of the 268 net additional requests, 54.9%. The direction is widespread and the size is concentrated. The counts are small, and Muse Spark 1.3's difference is two evaluations, which is not evidence of relative safety.

5.4 Individual ranks are not preserved

A positive correlation does not mean a given model keeps its position. On the Level 4 cohort, at +0.913:

- Claude Haiku 4.5: 473 / 645 (lowest) → 459 / 645 (second-lowest)
- Muse Spark 1.3: 481 / 645 (second-lowest) → 450 / 645 (lowest)

The two swap. We therefore claim positive association between rankings and nothing about the durability of a particular rank, a model-selection guarantee, or a predictive advantage of rankings over rates. No held-out deployment prediction was tested.

6 Matched decisions

A correlation and a total both aggregate. The matched design supports a third view: for each model and prompt, what happened under each track.

Level	Pairs	Both	Final-turn only	Source-turn only	Neither	Disagree	Net
L1	7,574	408	258	790	6,118	1,048	+532
L2	9,660	26	13	281	9,340	294	+268
L4	9,030	6,981	212	143	1,694	355	−69
L5	1,792	236	43	537	976	580	+494

Table 4: Matched outcomes per model and prompt, complete cases.

Level 4 is the case worth naming. Its aggregate barely moves, −69, but **355 pairs disagree**, and they disagree in both directions: 212 request only under the final-turn track and 143 only under the source-turn track. The near-equal total is the sum of offsetting changes, not an absence of change.

Reporting only the aggregate at Level 4 would omit the 355 observed pairwise disagreements. Recorded request presence differed between tracks in 355 of 9,030 matched pairs, 3.9%.

We do not attribute those differences to conversational history. This design does not separate delivery effects from repeated-execution variability: Section 7.1 shows disagreement between executions that begin with the same source user message.

7 Controls and sensitivity

7.1 Identical initial source input, separate executions

Where a source attempt has a single user turn, both tracks receive the same initial source message, configured system prompt and tool schema. They are separate executions, and they disagree:

Level	Cohort pairs	Disagreeing	All available pairs	Disagreeing
L1	2,506	318	3,015	338
L2	1,484	2	2,460	7
L3	1,120	0	1,294	1
L4	6,132	131	11,749	260
L5	140	5	3,469	34

Table 5: Single-user-turn sources: identical initial source input, separate executions. The first pair of columns uses the same complete-case cohort as Tables 1 to 4; the second uses every available matched pair. Both are correct and they answer different questions, so we give both rather than quoting one as the other.

The same initial source input, executed twice, produced different recorded outcomes in 131 of 6,132 Level 4 cohort pairs. Tool histories contain timestamps generated at execution time, and some provider settings use defaults, so subsequent model inputs and effective settings are not guaranteed identical. This stratum shows disagreement without a difference in source-user-turn delivery, but does not isolate model sampling from other execution variation. It does not quantify how much of Table 4's disagreement is attributable to either source, and we do not claim a general lower bound. The Level 3 cohort's zero disagreements in 1,120 pairs span 30 sessions, giving a 9.50% session-event upper bound.

7.2 Restricting to multi-user-turn sources

Level	Prompts	Sessions	Final-user-turn	Capped source-user-turn	Spearman	95% interval
L1	362	291	203 / 5,068	737 / 5,068	+0.911	[+0.81, +0.96]
L2	584	233	30 / 8,176	298 / 8,176	+0.607	[+0.39, +0.76]
L4	207	79	1,480 / 2,898	1,432 / 2,898	+0.680	[+0.47, +0.83]
L5	118	61	247 / 1,652	744 / 1,652	+0.964	[+0.84, +0.98]

Table 6: Restricted to source attempts with more than one user turn.

Level 4 attenuates most, from +0.913 to +0.680, and Level 2 from +0.673 to +0.607. Levels 1 and 5 barely move, at +0.911 and +0.964. The attenuation is therefore not uniform, and any statement of a restricted range must name its levels.

The Level 2 rate result survives the restriction and strengthens: 30 of 8,176 (0.37%) against 298 of 8,176 (3.64%). We report that as a sensitivity check, not as a replacement headline.

7.3 Selection and coverage

Complete-case selection equalizes the support across models; it does not make completion selection disappear. Level 5 retains 128 of 853 corpus prompts and Level 4 retains 645 of 1,293. Models whose multi-turn coverage was partial at the freeze contribute fewer prompts to the intersection, and the intersection is the prompts that completed everywhere.

Two further checks bear on whether the headline rests on a small part of the corpus.

Level	Difference	Largest single-session swing	Excluding input-blocked	Input-blocked evaluations
L2	+2.77 pp over 243 sessions	0.409 pp	+2.79 pp over 9,617 pairs	45
L4	−0.76 pp over 264 sessions	0.198 pp	−0.75 pp over 9,026 pairs	5

Table 7: Sensitivity of the paired rate difference.

Leave one source session out. Dropping each of the 243 Level 2 sessions in turn and recomputing, the largest single-session swing is 0.409 percentage points against a difference of 2.77, so no session carries the result. At Level 4 the largest swing is 0.198 points.

Excluding input-blocked rows. Those rows are retained in the denominators as evaluations with no request, and they are unevenly distributed across models. Removing them moves the Level 2 difference by 0.012 points and the Level 4 difference by 0.011. The treatment does not drive either figure.

8 Implications and limitations

8.1 What this supports

An evaluation table reports an ordering and a set of rates, and a reader takes one or the other depending on the decision. On this corpus those responded differently to one procedural change, and a stable total concealed movement in individual decisions. The practical form is a question to ask of any agent evaluation: which quantity does this claim rest on, and what would change it?

8.2 Relation to Paper 2

At Levels 2 to 4, Paper 2 [2] reports 0 registered transfers to unpermitted recipients in 69,297 evaluations behind a deterministic authorization check, a one-sided 95% upper bound of approximately 0.38% per source session over 790 sessions, while 25,370 evaluations still contained a successful simulated payment. This is the registered recipient endpoint, not a test of every policy rule or requester entitlement. Read together, model selection informs expected upstream behavior, with the caveats above, and authorization separately determines which requested actions execute. This paper does not test the second claim.

8.3 Limitations

Post hoc. Developed from an existing benchmark, not pre-registered.

Complete-case cohorts. Smaller and selected by completion; Level 5 is 128 prompts from 65 sessions.

Fourteen points per correlation. Coefficients over fourteen models are noisy; the intervals are marginal and condition on these models and recorded evaluations.

Single run per cell. No within-cell variance component; Section 7.1 shows run-to-run disagreement is non-zero.

Replay is not an adaptive attacker. Capped turns, a stopping rule after a successful payment, and omitted text-only assistant history. Differences between tracks are not a causal effect of conversational history.

No controlled cross-configuration contrast. Each attack exists at one level.

Level 3 has no ranking. One track is constant at zero.

Provider settings are not normalized, and 268 model-alone evaluations were recorded as input-blocked and retained as no-request records. The exception handler does not establish the blocking stage or recover any earlier activity in those rows.

Requests are not compromises, and permitted recipients do not establish legitimacy.

The refusal result is about this instrument. It shows the stored flag duplicates request presence here, not that refusal measures generally carry no information.

9 Reproducibility

Every table in this paper can be recalculated from the outcome records and source-turn metadata in the dataset package [1], without internal paths or new model calls. Public gated access is scheduled for September 21, 2026. From the root of the dataset:

```
python3 reproduce/rank_stability.py
```

It reads `outcomes/outcomes.parquet` and `corpus/public-corpus.jsonl`, both included in the release, and requires `pyarrow`. It rejects duplicate evaluation keys, warns if the model roster is not the fourteen reported here, and exits rather than treating a prompt with no recorded

turn count as single-turn. The seed is 20260915 and the resample count is 1,500, as in Section 4.3.

The source is snapshot `frozen-20260910T1230Z`, copied on September 10, 2026 at 12:30 UTC. Model-alone records with status `ok` have attempt timestamps from September 4 at 23:39:28.258 UTC through September 10 at 12:28:24.744 UTC. The later September 6 start in the run manifest is resume metadata, not the first evaluation. The package's `methodology/run-manifest.json` records all fourteen model identifiers; `methodology/PRE_REGISTRATION.md`, its addenda and `levels/` record the configurations. Many identifiers are provider aliases, not immutable model versions. Generation seeds and provider-returned model revisions were not recorded, and the manifest does not pin the executed source commit. Current implementation code includes a September 16 Gemini tool-response correction after the freeze; it must not be treated as the exact implementation that generated every recorded call.

The dataset package is at `huggingface.co/datasets/aporthq/vault-benchmark-v1`; the analysis artifact is pinned to revision `a1669bd77071239df57910f43e0667ee35518bde`. Its outcome records include evaluations whose transcripts are withheld under provider terms. The original replay implementation, judge prompts and raw archive are not part of the public package. The archive's judge panel is Mistral Medium 3.5 and Grok 4.6; no judge labels are used to calculate this paper's request endpoint. Readers can recalculate the reported summaries, but we do not promise identical reruns of the original provider calls. `release/rank_stability.py` separately reproduces the analysis from the internal frozen snapshot.

CC BY 4.0 covers APort's compilation and the APort-authored materials identified in the dataset card, not a blanket licence over participant prompts or provider outputs. Their separate terms are stated in the card; this paper does not assign a software licence to the analysis scripts.

10 Conclusion

Fourteen models, 4,371 recorded human attacks, and two replay procedures. The ordering of the models is positively associated across tracks where defined, at +0.673 to +0.967, and at +0.607 to +0.680 at Levels 2 and 4 once single-user-turn sources are excluded. Request rates differ by configuration: 39 to 307 of 9,660 at Level 2, nearly eightfold, while Level 4 changes from 7,193 to 7,124 of 9,030, with 355 disagreeing pairs beneath that net difference of 69.

Rankings, rates and individual decisions are three different things an evaluation reports at once. Saying which one a claim rests on, and what perturbation it survives, costs nothing.

11 Disclosure

The author is the founder of APort Technologies Inc., which develops the authorization layer evaluated in the companion paper [2]. This paper reports model behavior with no authorization layer present and makes no claim about that product. The benchmark was designed, run and analyzed by the author. The planned release includes all recorded evaluation outcomes, including counterexamples to these conclusions; raw records and specified transcripts are withheld as described in Section 9.

Data terms. The CTF terms in effect state that attempt logs are anonymized and used for security research. They do not grant a CC BY licence over participant prompts, and the dataset card licenses APort's compilation rather than the prompts themselves. We state the documented terms rather than a broader consent.

All transfers evaluated in this paper were simulated; no real funds moved through the evaluated payment tool. Release processing uses redaction and identifier transformation, followed by filtering for residual identifying patterns. These checks do not guarantee anonymity: identifying text or linkage may remain. The dataset card describes the removal-request process.

References

[1] U. Uchibeke. APort Vault benchmark, v1. huggingface.co/datasets/aporthq/vault-benchmark-v1, 2026.

[2] U. Uchibeke. APort Vault: Benchmarking AI Agent Payment Authorization with the Open Agent Passport. Unpublished companion manuscript, 2026; public release scheduled for September 21.

[3] U. Uchibeke. Before the Tool Call: Deterministic Pre-Action Authorization for Autonomous AI Agents. arXiv:2603.20953, 2026.

[4] M. Andriushchenko, A. Souly, M. Dziemian, et al. AgentHarm: A Benchmark for Measuring Harmfulness of LLM Agents. arXiv:2410.09024, 2024.

[5] E. Debenedetti, J. Zhang, M. Balunović, L. Beurer-Kellner, M. Fischer, and F. Tramèr. AgentDojo: A Dynamic Environment to Evaluate Prompt Injection Attacks and Defenses for LLM Agents. arXiv:2406.13352, 2024.

[6] A. Souly, Q. Lu, D. Bowen, et al. A StrongREJECT for Empty Jailbreaks. arXiv:2402.10260, 2024.

[7] P. Chao, E. Debenedetti, A. Robey, et al. JailbreakBench: An Open Robustness Benchmark for Jailbreaking Large Language Models. arXiv:2404.01318, 2024.

[8] Y. Gao. How Reliable Is Your Jailbreak Judge? Calibration and Adversarial Robustness of Automated ASR Scoring. arXiv:2606.25487, 2026.

[9] Z. C. Lipton and J. Steinhardt. Troubling Trends in Machine Learning Scholarship. arXiv:1807.03341, 2018.

[10] M. Sclar, Y. Choi, Y. Tsvetkov, and A. Suhr. Quantifying Language Models' Sensitivity to Spurious Features in Prompt Design, or: How I Learned to Start Worrying about Prompt Formatting. ICLR 2024. arXiv:2310.11324.

[11] A. Hua, K. Tang, C. Gu, J. Gu, E. Wong, and Y. Qin. Flaw or Artifact? Rethinking Prompt Sensitivity in Evaluating LLMs. EMNLP 2025.

[12] X. Bouthillier, P. Delaunay, M. Bronzi, et al. Accounting for Variance in Machine Learning Benchmarks. MLSys 2021. arXiv:2103.03098.

[13] P. Liang, R. Bommasani, T. Lee, et al. Holistic Evaluation of Language Models. TMLR 2023. arXiv:2211.09110.

[14] U. Uchibeke, K. Schneider, S. Hosseinzadeh Kassani, and R. Deters. Blockchain Access Control Ecosystem for Big Data Security. IEEE Cybermatics, 2018. arXiv:1810.04607.