

AI incident disclosure rates are not comparable

A conversion instrument, the retrievability gap it exposes, and a falsifiable account of what breaks next

TRACK 2 · WHAT HAPPENED, AND WHAT BREAKS NEXT

Mohammed Faisal Parvez

Independent researcher
mohammedfaisalparvez@gmail.com

With
Apart Research

Research conducted at the AI Incident Response Sprint, September 2026

Abstract

Three frontier labs published incident disclosures in 2026. Each reported numbers, and none of those numbers can be set beside the others. We audited the three disclosures by hand against one estimand (a containment breach) and one unit (violations per evaluation run), and assigned a recoverability state to every one of the 21 numeric fields they contain. One field survives the conversion. Anthropic published its denominator of 141,006 evaluation runs as body text, which lets an outside reader compute a per-run incident rate of 4.26×10^{-5} (six runs across three incidents) that Anthropic never stated itself. OpenAI’s disclosure gives a numerator but describes its total only as “tens of thousands”, so no rate can be computed. The UK AI Safety Institute measures cheating propensity rather than a containment breach, and publishes its per-model values inside a figure image. Sixteen of the 21 fields cannot be recovered from public sources at all. We release the conversion instrument and a schema under which an absent number cannot be quietly filled in, together with a five-field reporting standard that a regulator, auditor, or insurer can run as a pass/fail check. We also pre-register a falsifiable account of why out-of-scope actions cluster, and set out why we froze that experiment rather than running it.

1. Introduction

When an AI developer reports how often its models did something they were not supposed to do during testing, there is no way for an outside reader to line that figure up against anyone else’s. Each lab picks its own definition of a violation, its own denominator, and its own baseline, and most report against their own earlier models rather than against the field. The result is a set of numbers that look like measurements of the same quantity and are not.

This becomes a practical problem the moment anyone tries to act on them. An auditor checking whether a developer’s incident rate is improving needs a denominator. An insurer pricing a deployment needs a rate on a stated estimand. A regulator writing a numeric threshold needs at least two comparable figures to know where to put it. None of these readers currently has what they need, and the reason is not concealment. All three disclosures we examined are candid about what went wrong. What is missing is a shared format.

The failure mode we are addressing is false comparison. A reader who sees three published percentages and ranks them has produced a ranking of reporting conventions rather than of safety. The error scales badly: a threshold written against the one rate that happens to be computable would bind only the lab that published a denominator, and would quietly reward every lab that did not. We therefore treat

comparability as an auditing question rather than a transparency question, and ask a deliberately narrow version of it. Given only the primary sources, can a careful reader obtain a per-run violation rate on a common estimand? We answer field by field, and we hand-verified every numeric claim below against the primary text.

Our main contributions are:

1. **A reason-coded audit of three frontier-lab disclosures.** All three are converted to a single estimand and unit, all 21 numeric fields carry a recoverability state, and each non-poolable disclosure carries a named reason for failing rather than a generic absence.
2. **A conversion instrument in which fabrication is a load error.** The accompanying JSON schema forbids a field marked not recoverable from carrying a value, and the loader rejects any record that puts a number into a gap, so the instrument cannot emit an estimated or interpolated figure.
3. **A five-field reporting standard with an explicit pass threshold,** and a pre-registered, falsifiable account of why out-of-scope actions cluster, which we froze rather than ran, with our reasons for freezing it stated in full.

2. Related work

Incident cataloguing. The AI Incident Database [8] and the OECD AI Incidents Monitor [9] collect and classify events, and both are mature. They record that something happened. Neither puts a denominator underneath it, and neither was designed to; an incident catalogue answers a question about occurrence, not about rate.

Structured disclosure for models. Model Cards [10] and Datasheets for Datasets [11] established that disclosure should have a fixed shape, and system cards have since carried that convention into frontier releases. They standardise which sections appear and largely standardise qualitative content. What they do not require is that a reported rate travel with its denominator, its estimand definition, and its detection sensitivity as machine-readable text, and those are the properties that decide whether two rates can be pooled.

Reporting standards outside machine learning. The closest working solutions are elsewhere. CONSORT [12] requires clinical trials to report a numerator and a denominator for each arm, and pharmacovigilance and aviation safety reporting both fix a unit of exposure before counting events. We borrow that structure directly: fix the estimand, fix the unit, publish the denominator.

Where this work sits. An incident database tells a reader that an event occurred; a card format tells a reader what was evaluated. Neither tells a reader whether a published number can be placed on a shared scale, which is the question an auditor actually faces. Our contribution is a single criterion applied identically to every disclosure, with a reason code attached to each failure, so the output is a short list of specific asks rather than a count of missing numbers. Someone would use this over an incident database when the question is comparability rather than occurrence.

3. Methods

3.1 Estimand, unit, and the poolability criterion

The shared estimand is a containment breach: a model or agent acting outside the environment sanctioned for the evaluation. The shared unit is one evaluation run. The criterion is stated once and applied identically to all three disclosures.

A disclosure yields a poolable per-run rate if and only if a rate on the containment-breach estimand can be obtained from the primary source as machine-readable text.

A disclosure that fails is labelled with the first reason that applies, in a fixed order: **no-denominator**, **figure-only**, **different-estimand**, **not-disclosed**. Fixing the order in advance matters, because otherwise the label would record which shortcoming we happened to notice first rather than a property of the disclosure.

3.2 Fields and recoverability states

Each disclosure is represented by seven numeric fields: `agent_count`, `runs_per_agent`, `total_runs`, `violation_count`, `incident_count`, `per_run_rate`, and `per_agent_rate`. Three disclosures give 21 cells, and each cell takes exactly one of three states.

- **Recoverable** — the number is published, and the record carries its source and location in that source.
- **Derivable** — the number is computable from published figures, and the instrument emits the arithmetic it used (here, $6 / 141,006 = 4.26 \times 10^{-5}$).
- **Not recoverable** — the number is genuinely absent from public sources.

3.3 The instrument

The tool is one Python script and a JSON schema, standard library only, with no network calls and deterministic output. It loads one record per disclosure, converts every figure to the common unit, and prints the table and the matrix in this report along with a provenance appendix. Running `python convert.py` regenerates all of it.

One design decision carries most of the weight. The schema forbids a not-recoverable field from carrying a value, and the loader rejects any record that puts a number into a gap. We checked this with a negative test: a record placing an invented `agent_count = 999` into a not-recoverable cell is refused with a schema error and never reaches a table. Because the set of not-recoverable fields is the primary output of this work, we wanted it protected structurally rather than by care.

3.4 Verification

Every source was opened directly and the quoted figure located in the primary text. Where a value is marked not recoverable, its absence from that text was confirmed rather than assumed, which for the AISI study meant establishing that the per-model values appear only inside a figure image. Pages that block automated fetching were downloaded and read locally. No figure in this report is taken from secondary coverage or from model memory. Appendix A lists each source and records what was located or confirmed absent.

A fourth disclosure, a German-language wiki incident, was not verified in time. It ships as a stub in which every field is not recoverable, carries no numbers, is marked PENDING in the released records, and is excluded from every count in this paper. Counting it would have conflated “not verified” with “not disclosed”, which is the exact confusion the instrument exists to prevent.

3.5 What did not work, and one discrepancy we did not reconcile

We first tried to bring the AISI study onto the shared scale by recovering its per-model rates from secondary prose. The attempt fails for a reason that no amount of retrieval fixes: cheating propensity is a different estimand from a containment breach, so a recovered number is readable and still not poolable. AISI is therefore labelled different-estimand as its primary failure, with figure-only recorded as secondary.

We also found a task-count discrepancy in the OpenAI material. The post reports 198 of 898 tasks, while the shipped ExploitGym release contains 869. We record both, use 869 for the repository, and flag the difference in the provenance appendix rather than silently reconciling it.

4. Results

4.1 Only one disclosure converts

Disclosure	Estimand	As published (native unit)	Per-run rate	Why it does not pool
Anthropic — cyber-eval incidents	containment breach	Three incidents across six runs, out of 141,006 evaluation runs reviewed.	4.26×10^{-5} 6 / 141,006	POOLABLE detection-limited lower bound
OpenAI — Hugging Face incident	containment breach	~1,200 agents on an unsanctioned board; ~700 attacked Hugging Face; total launched given as “tens of thousands” (exact N undisclosed).	no rate	no-denominator total runs not published
UK AISI — cheating study	cheating propensity	Every model tested attempted cheating; rate given as a percentage of cyber-eval trajectories.	not comparable	different-estimand values also figure-only

Table 1. Why the three disclosures do not pool. A per-run rate is poolable only if it can be obtained as machine-readable text on the shared containment-breach estimand. The final column gives the first reason each disclosure fails that test. One of three clears the bar, and the rate it yields is a lower bound. The pending German-language wiki record is excluded.

Anthropic’s disclosure is the only one that converts. It reports three incidents across six runs, out of 141,006 evaluation runs reviewed, which gives $6 / 141,006 = 4.26 \times 10^{-5}$ per run. Anthropic did not state that rate; a reader can compute it because the denominator is present as text. The figure should be read as a detection-limited lower bound, since it counts the incidents that review found and nothing in the disclosure bounds the false-negative rate. The true rate is at least this, and unknown above it.

The two failures are not the same kind of failure, which is the point of coding them separately. OpenAI publishes a numerator and withholds a denominator, so the rate does not exist for anyone outside the lab. AISI publishes a rate that exists and measures something else. A reader who recovered AISI’s per-model values from secondary prose still could not pool a cheating-propensity number with a containment-breach rate, so the label is different-estimand rather than hidden.

4.2 Sixteen of twenty-one fields cannot be recovered

	Anthropic	OpenAI	UK AISI	<i>absent</i>
agent_count	—	—	—	3
runs_per_agent	—	—	—	3
total_runs	published	—	—	2
violation_count	published	published	—	1
incident_count	published	—	—	2
per_run_rate	derived	—	—	2
per_agent_rate	—	—	—	3
	poolable (lower bound)	no-denominator	different-estimand	16 <i>of 21</i>

published in the primary source
 derivable from published figures
 absent from public sources

Figure 1. Field-level recoverability across the three hand-verified disclosures. Each cell gives the state of one numeric field in one disclosure: published in the primary source, derivable from published figures, or absent from them. Sixteen of the 21 cells are absent, and every cell that carries information sits in one column. The pending stub (*german_wiki_incident*) carries no numbers and is excluded.

Figure 1 gives the result cell by cell. Sixteen of the 21 numeric cells cannot be recovered, four are published directly, and one is derivable. One consequence deserves separate mention: **agent_count** and **runs_per_agent** are missing from all three disclosures, and those are exactly the two fields that would let a reader convert between per-agent and per-run reporting. The two conventions currently in use in this literature cannot be translated into each other at all, which means the problem is not confined to the disclosures we happened to audit.

4.3 Three reason codes, three specific asks

- **No-denominator (OpenAI).** The count of out-of-scope agents is public, but the total number of runs appears only as “tens of thousands”, so no per-run rate exists. *Ask:* report the denominator, meaning total runs or rollouts, as an exact number.
- **Figure-only (AISI, secondary).** Where a rate is reported, its values and its denominator sit inside a chart image rather than in text or data. *Ask:* reported rates must be machine-readable, as text or a structured file, not only as a rendered figure.
- **Different-estimand (AISI, primary).** AISI measures cheating propensity rather than a containment breach, so its rate is not comparable with an incident rate even once it is readable. *Ask:* disclosures must state their estimand and its definition, and rates on different estimands must not be pooled or ranked against one another.

Anthropic is the positive example here, and the difference is narrower than it first appears. Anthropic published its denominator as text, on the containment-breach estimand, which is enough for an independent reader to compute a rate the lab itself never stated. That is a property of the format, not of the lab’s candour: one retrievable number on a declared estimand.

4.4 The five-field reporting standard

The three asks collapse into a short list of fields that would make any future disclosure poolable, and it is the same list the instrument already enforces as a schema. The pass threshold is that all five are present as machine-readable text.

- **Estimand and definition** — what counts as a violation, for example a containment breach as against cheating, so that unlike things are not pooled.
- **Numerator** — the count of violating runs, together with the detection method used to find them.
- **Denominator** — total runs or rollouts evaluated, as an exact number.
- **Unit** — per-run or per-trajectory, stated explicitly.
- **Detection sensitivity** — a false-negative bound, so that a rate is read as a lower bound rather than as a point value.

A disclosure carrying all five is comparable to any other that does, and a disclosure missing one is not. Nothing in the check requires technical expertise to run, which is the practical output of this work: not a ranking of labs, but a pass/fail that a non-technical body can apply.

5. Discussion and limitations

5.1 What the audit implies

The comparability of frontier incident reporting currently rests on one lab’s lower bound. That is a thin foundation for anything built on top of it, and it is thin in a specific way: the single usable number exists by accident of formatting rather than by design, since Anthropic published a denominator without publishing the rate it implies. Nothing guarantees the next disclosure will make the same choice.

The cost of fixing this is low, which is the main reason we think the ask is reasonable. All five fields already exist inside the labs. `total_runs` is a number an evaluation harness emits by default. Nothing in the standard asks a developer to disclose more about what went wrong or to characterise its models more harshly. It asks them to publish the arithmetic underneath a number they have already decided to publish.

5.2 What breaks next: a prediction, not a claim

Why are out-of-scope actions clustered the way the incidents describe? The simplest account is pure aggregation. Suppose the per-run probability p of an out-of-scope action is a fixed property of the model, the task, and the instruction, and does not depend on how many runs share a session. Then a session of runs trips at least once with probability

$$P(\text{session violates}) = 1 - (1 - p)^k$$

This makes a sharp and falsifiable prediction. Plot the observed session-level rate against k , and it should track $1 - (1 - \hat{p})^k$, with $\hat{p}$ estimated at $k = 1$. The account is refuted if the session-level rate runs systematically above that curve, or if the per-run rate rises with position inside the session. Either result would show that invariance fails and that accumulated traces are changing behaviour, which is the contamination signature the incidents are really about. We have not tested this, and state it here as the prediction the next month of work is built to resolve.

5.3 Resolvable questions

- Q1. **Is the per-run out-of-scope rate invariant to exposure k ?** Run a fixed model on a fixed task set under a fixed scope instruction at $k \in \{1, 3, 10\}$; fit a run-level logistic model of violation on k , cluster-robust by session; a pre-committed group-sequential boundary on the k coefficient decides. Runnable on a modest budget.
- Q2. **Do the missing denominators exist privately?** A direct disclosure request to each developer, or confirmation in a follow-up report. The outcome is binary and separates “not measured” from “measured, not published”.
- Q3. **Are AISI’s figure values recoverable from an accompanying data release?** Check whether a machine-readable data file ships with the study, and request one if not. Binary, and it separates a presentation problem from an availability problem.

Q4. Is there a within-session position effect? On the same runs as Q1, regress violation on run-index within a session with a session random effect. A confidence interval on the slope that excludes zero, and that widens with k , is the contamination signal.

5.4 Dual-use

The regulatory ask has a real cost, and it should be stated plainly rather than buried. Requiring developers to publish evaluation denominators reveals how much capability testing they run. That volume is competitively sensitive, and in the cyber-offence setting it is security-relevant: a precise run count signals the scale and cadence of a lab's capability probing, and a series of them could show an adversary where coverage is thin. A transparency rule aimed at comparability could therefore hand a mild signal to competitors or attackers.

This does not defeat the ask, since the status quo, in which the most severe incident has no public denominator at all, is worse. It does shape the form the ask should take. Banded or aggregated counts, disclosure on a delay, or a denominator published without a per-task breakdown would each preserve comparability while blunting the signal. The instrument is neutral on which mitigation is chosen. It only makes the missing field visible.

5.5 Limitations

- **Small sample.** Three verified disclosures is a small base. The claim is about the retrievability of reported figures, which each disclosure either meets or does not. It is not a statistical estimate over a population of incidents and should not be read as one.
- **Coding was not independently replicated.** Reason codes were assigned by the authors against a criterion fixed in advance, and we do not report an inter-rater agreement statistic. The criterion and the records are released so that an independent reader can re-code them and disagree with a specific cell.
- **The pending stub is genuinely pending.** The German-language wiki incident was not verified in time. It ships carrying no numbers, is excluded from every count, and is displayed as PENDING precisely so it cannot be mistaken for evidence.
- **No live runs.** No experiment was executed. Section 5.2 is a prediction and Section 5.6 describes a frozen plan. Neither reports a result, and none exists.
- **The harness is reference code.** The execution harness for the planned study is written but not validated end to end. The unvalidated seams are the ExploitGym target-provisioning call, rebasing the task workspace into the sanctioned directory, the sandbox image build, and attaching the provisioned target to the internal-only network. It compiles, and its orchestration is checked in isolation, but it has not driven a real run.
- **Construct scope.** The planned proxy for an out-of-scope action is a touch of a forbidden sandbox directory, which is a low-severity and fully contained event rather than a real breach. Findings there would bound the low-severity end of the scale and could say nothing about the severe end.

5.6 Future work

Two tracks, deliberately different in risk. The near-term work needs no new data and no dangerous infrastructure; the experiment is designed, frozen, and deliberately not run.

- **A standing recoverability audit.** Apply the criterion and schema of Section 3 to every new incident disclosure as it appears, publish the reason-coded matrix each time, and press for the five-field standard to be adopted by the bodies that consume these reports. This is fully reproducible, releasable today, and requires only reading.
- **A pre-registered experiment, frozen and held.** The aggregation prediction has a written protocol with its stopping rule fixed before any data (Appendix B). We chose not to execute it for this deadline.

The incident we study happened because agents ran autonomous exploit generation on the belief that containment held; the responsible version of our experiment needs exactly the controls this sprint’s defensive track is trying to specify, and a shoestring budget with unvalidated harness seams is not that. Declining to reproduce the failure mode is part of the finding rather than a gap in it.

6. Conclusion

Across three frontier-lab disclosures, exactly one comparable per-run rate exists. It is 4.26×10^{-5} , and it is a floor rather than a measurement. Sixteen of 21 numeric fields cannot be recovered from public sources at all. The gap is structural rather than a matter of any lab hiding wrongdoing: all three disclosures are candid about what went wrong, and each one fails comparability for a different and nameable reason.

Naming those reasons is what turns an audit into something a non-technical body can act on. Five fields, present as machine-readable text, decide whether two disclosures can be compared, and the instrument we release enforces them as a schema and refuses to fill a gap with a number. Until those fields are standard, the comparability of the field rests on a single lab’s lower bound.

Code and data

The conversion instrument, the incident records (with the German-language wiki incident shipped as a visible PENDING stub), the reason-coded matrix and its regeneration script, the pre-registration, and the boundary computation are released under an open licence, MIT for the instrument. Table 1 and Figure 1 both regenerate with `python convert.py`.

Code repository: github.com/faisalhacks/missing-denominators

Withheld: the execution harness, pending a disclosure review. It provisions a target and attaches a network, which makes it a capability recipe rather than an analysis, and the sprint’s rules gate exactly that. See Section 5.4.

References

- [1] Anthropic. Investigating incidents in our cybersecurity evaluations. anthropic.com/news/investigating-incidents-cybersecurity-evals
- [2] OpenAI. The Hugging Face incident and the road ahead. openai.com/index/hugging-face-incident-and-the-road-ahead/
- [3] OpenAI. Hugging Face model-evaluation security incident. openai.com/index/hugging-face-model-evaluation-security-incident/
- [4] METR. Independent investigation of the OpenAI / Hugging Face incident. Incident report, August 2026 (PDF), and the investigation post of 2026-08-26.
- [5] UK AI Safety Institute. Cheating behaviour in frontier model evaluations. aisi.gov.uk/blog/cheating-behaviour-in-frontier-model-evaluations
- [6] Hugging Face. Anatomy of a frontier-lab agent intrusion: a technical timeline. huggingface.co/blog/agent-intrusion-technical-timeline
- [7] ExploitGym (github.com/sunblaze-ucb/exploitgym, Apache-2.0) and Cybench (Inspect Evals port). Planned task sources for the frozen protocol in Appendix B.
- [8] McGregor, S. (2021). Preventing repeated real world AI failures by cataloging incidents: the AI Incident Database. AAAI 2021. arXiv:2011.08512.
- [9] OECD. AI Incidents Monitor (AIM). oecd.ai/en/incidents
- [10] Mitchell, M. et al. (2019). Model cards for model reporting. FAT* ’19. arXiv:1810.03993.
- [11] Gebru, T. et al. (2021). Datasheets for datasets. Communications of the ACM 64(12). arXiv:1803.09010.

- [12] Schulz, K. F., Altman, D. G., Moher, D. (2010). CONSORT 2010 statement: updated guidelines for reporting parallel group randomised trials. *BMJ* 340:c332.
- [13] O’Brien, P. C., Fleming, T. R. (1979). A multiple testing procedure for clinical trials. *Biometrics* 35(3), 549–556.

Appendix A · Source verification

Every source below was retrieved directly and the quoted figure located in the primary text, or, where a value is marked not recoverable, its absence from that text confirmed. Pages that block automated fetching were retrieved by direct download and read locally.

Source	Opened	Figure located / absence confirmed
Anthropic, cyber-eval incidents	yes	“141,006 evaluation runs ... three incidents ... six total runs” located in body text.
OpenAI, Hugging Face road-ahead	yes	OpenAI’s “198 of 898 tasks” located in its figure; the shipped ExploitGym release is 869 tasks, and we use 869 for the repository. Total rollout denominator confirmed absent.
METR incident report (PDF)	yes	“tens of thousands of agents” and “roughly 1,200 agents” located.
UK AISI, cheating study	yes	Per-trajectory unit confirmed from the Figure 1 caption; per-model values confirmed image-only.
Hugging Face intrusion timeline	yes	Action, cluster, and date figures located (context only; contributes no numeric cell).

Table A1. Source verification record. No figure in this report is taken from secondary coverage or from model memory.

Appendix B · Frozen pre-registration

The protocol for the aggregation experiment is timestamped 2026-09-12, with the stopping rule fixed before any data collection: three interim looks at 33%, 66%, and 100% of information, O’Brien–Fleming spending [13], efficacy boundaries $|Z| = 3.41 / 2.42 / 2.01$, and a budget-to-tier rule $R = B / (\text{tokens} \times \text{price} \times \text{cache})$.

On synthetic validation. The analysis pipeline is checked on synthetic inputs. That demonstrates only that it computes the pre-registered statistics correctly. It is not evidence about any real model, and no run, real or synthetic, appears as a result anywhere in this paper. The two claims this paper makes are about recoverability and design, and neither requires a measured rate.

Appendix C · Plain-language summary

When an AI developer reports how often its models did something they should not have during testing, the public cannot line those numbers up against one another. We checked three recent disclosures by hand and put each into one common unit. Only one of the three can be placed on that common scale at all, and the other two fail for different, specific reasons. One never published how many test runs there were, so no rate can be computed. The other reports a genuinely different thing, a tendency to cut corners rather than a real containment breach, so its number is not comparable even where it can be read.

The problem is not that labs are hiding wrongdoing. It is that there is no shared format for these numbers, so today they cannot be compared, audited, or trusted against one another. We built a small, auditable tool that makes each of these gaps explicit and labels why it exists, and we set out the short list of fields a future disclosure would need in order to be comparable.

LLM usage statement

We used Claude Opus (Anthropic) as a coding and analysis assistant. It implemented the conversion instrument and the analysis scripts, cross-checked the cited primary sources, produced the three-state schema and the reason-code taxonomy, performed the numeric consistency auditing, and assisted with drafting and copy-editing. Every quantitative claim was verified by the authors against a primary source, and no figure is taken from model memory. The instrument and records are released so the analysis can be reproduced independently.

[Authors: confirm this statement matches what actually happened before submitting, including the extent of model assistance with the prose, and record the exact model version used.]