AUDITA: certified auditing and causal attribution of adverse outcomes in autonomous multi-agent systems

Zhixu Du[1,**], **Yiran Chen**[1]

[1]**Department of Electrical and Computer Engineering, Duke University, Durham, North Carolina, USA**

Abstract

Physical automation is scaling toward fleets of embodied machines commanded by an AI brain. Early deployments already run factories and warehouses at production rates beyond any human line, and their adoption is accelerating. But when their joint decisions cause harm, everyone involved has reason to blame everyone else, the machine vendor, the algorithm provider, the factory operator, the insurer, and the regulator, and no method can divide the responsibility between them. Existing methods read logs whose origin they cannot verify and name a single culprit, misrepresenting outcomes that are overdetermined, preempted, or caused by an omission. We present AUDITA, an audit layer pairing a tamper-evident record of every inter-agent command with a certified, graded causal-attribution engine. We prove its verdict cannot be gamed: a rule-following agent can never be made to look guilty, an attempt to shift blame is itself caught and graded, and we establish the exact limit of what an evidence-based auditor can certify. On live language-model pipelines it reduces the standard judge baseline's responsibility error roughly threefold; on a benchmark of accident-grounded structures it recovers responsibility where single-culprit baselines fail, and stays invariant under forgery. AUDITA turns the question of who is to blame from an argument about logs into a calculation over evidence.

1. Introduction

The coming operating model for physical automation is an AI brain commanding fleets of embodied machines. Language-model planners already ground instructions into robot actions[1,2], vision-language-action models put general-purpose control onto physical platforms[3,4], foundation models target one brain commanding many bodies[5], and orchestration frameworks wire specialised agents into pipelines that plan, delegate, execute, and verify[6,7]. As these collectives enter fabrication cells, warehouses, and shared public spaces, their joint decisions carry physical and economic consequence: a mistimed handoff can injure a worker, and a hallucinated instruction can propagate through three delegations before any actuator moves.

When such an adverse outcome occurs, a specific question follows, asked by the operator deciding what to fix, the insurer pricing the loss, the regulator demanding an account, and the vendor whose component stands accused: *which agents, through which commands, bear how much of the responsibility, and which are demonstrably innocent?* Today that question has no defensible answer. Execution logs exist for debugging. Nothing in them proves origin, order, or completeness once liability is at stake. Regulation sharpens the demand into an obligation: the European Union's AI Act requires high-risk systems to record events so that operation can be traced and audited[8], and collaborative-robot standards presume incidents can be reconstructed[9,10]. What the obligation lacks is a mechanism.

*Correspondence E-mail: zhixu.du@duke.edu

Research attention has recently turned to the attribution half of the problem. The Who&When benchmark formalised failure attribution for multi-agent systems and found the best judge-style methods reach only 53.5% agent-level and 14.2% step-level accuracy[11]; a taxonomy showed that much failure lies in inter-agent coordination rather than any single model[12]. Trained attributors inject and replay faults[13], step-level counterfactual scoring converts failures into validated repairs[14], and replay engines validate an estimator against synthetic ground truth[15]. Concurrent work argues that agent systems cannot be accountable without an auditability layer and maps its requirement dimensions[16] (a full survey is in the Supplementary Information). Beneath this progress sit two structural absences.

First, *an evidence layer*. Every method operates on logs it must take on faith: nothing binds authorship, order, or completeness, so the output inherits the corruptibility of its input, and in any setting with liability at stake the records *will* be contested. Second, *causal semantics that match how blame behaves*: attribution targets a single culprit, but adverse outcomes are routinely overdetermined (two commands, either sufficient), preempted (the blamed command's effect never arrived), or caused by omission (a supervisor who never issued the required halt), and responsibility is then a matter of degree, not a pointer. A century of legal doctrine[17,18] and two decades of formal work on actual causality[19–21] treat exactly these structures; the attribution literature for agent systems has yet to absorb them.

We present AUDITA (Figure **1**), an audit layer that closes both gaps and couples them, so the causal analysis is computed only over evidence that survives verification. Every inter-agent message is signed by its author, cites the messages it acts on, and is sealed into an append-only, Merkle-committed record[22–24], which makes authorship, order, and completeness cryptographically checkable. Any declared predicate over the world or the work product triggers the audit, so the same machinery handles an injury and a ruined batch. A production-gated causal test scores each principal by counterfactual dependence under the modified Halpern–Pearl definition[20], returning a responsibility degree $\rho \in [0, 1]$[21] that handles redundancy, preemption, and omission natively; every reported cause is re-executed under intervention[25], so claims arrive with verification receipts. A negligence-structured verdict separates duty, breach, causation, and harm[17], and reports a graded, evidence-linked account intended as an *input* to human and legal judgment, not a substitute.

The composition carries a package of guarantees we prove under three explicit, standard assumptions (Methods): a replayable stack, a recorded channel, and sound key custody, meaning keys are never stolen while the insiders holding them may be arbitrarily malicious. Involvement can be manufactured against a compliant principal, but culpability cannot, because every culpability finding is witnessed by the accused's own certified conduct violating a declared duty (Theorem 1); a coalition that shifts blame onto a compliant principal is itself named and graded by a machine-checkable certificate (Theorem 2), a guarantee that persists at every meta-order (Theorem 5); destroying evidence cannot silently exonerate a certified culprit (Theorem 3); and a completeness barrier bounds what any record-confined auditor can certify (Theorem 4).

We evaluate AUDITA under a pre-registered protocol on two registers: live language-model multi-agent pipelines solving GSM8K[26], where it cuts judge attribution error roughly threefold; and a benchmark of physical incident structures grounded in public robot-accident records, where it recovers planted responsibility exactly, refuses to blame anyone on genuine accidents where judges always name someone, and holds its verdict invariant under record forgery. AUDITA audits the *event*, not the model: it answers, for a concrete adverse outcome, who did what, whether it mattered, and to what degree, with evidence that would survive a hostile audience. We argue this layer is a precondition for autonomous collectives operating anywhere accountability is demanded.

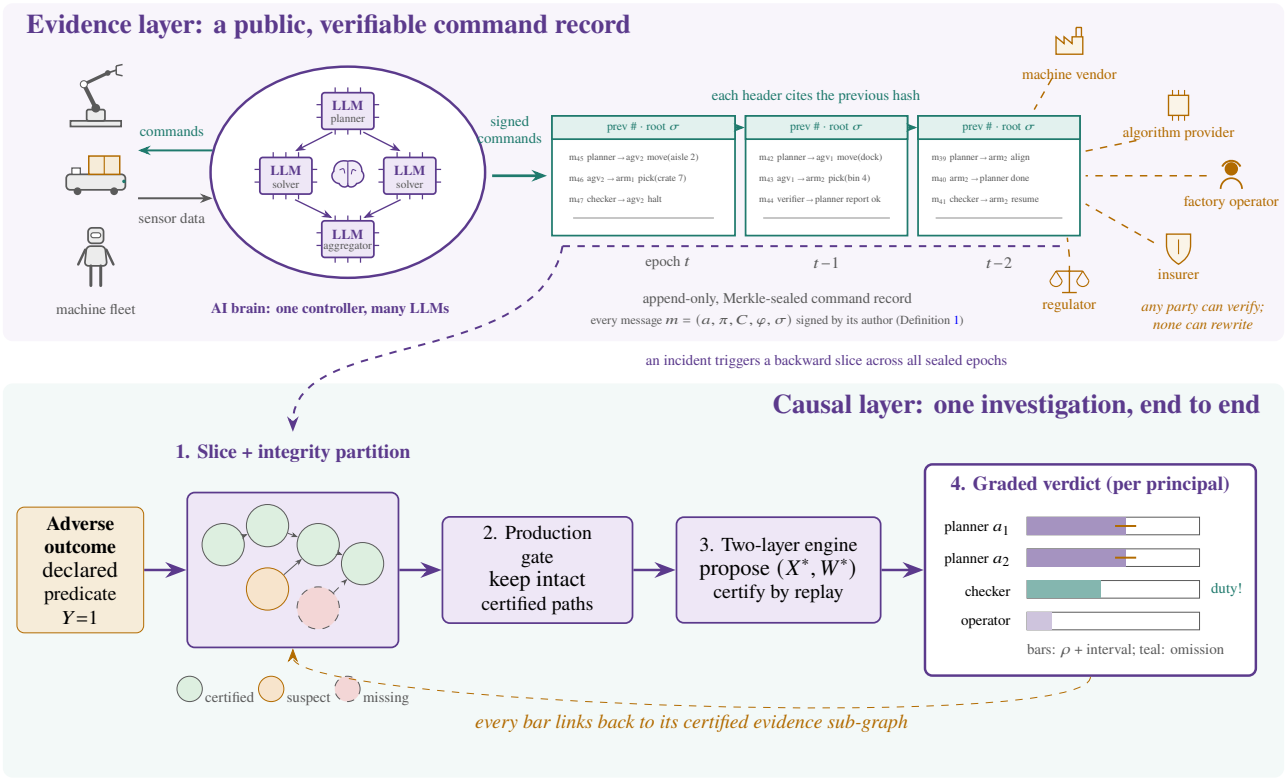


Figure 1 | AUDITA's two layers. *Top, evidence layer.* An AI brain, one controller running many language models (a planner, parallel solvers, an aggregator), commands a machine fleet and receives its sensor data. Every inter-agent command is a signed message $m = (a, \pi, C, \varphi, \sigma)$, author, payload, citations, committed effect, signature, appended with its delivery receipt to a public record in the blockchain sense (Definition 1). An *epoch* is a fixed window of this message stream: at its close the recorder signs the Merkle root over the window's messages and chains it to the previous block's hash, so the record is append-only and tamper-evident. The blocks show illustrative episodes of one shift (agv, an automated guided vehicle; arm, a manipulator arm): in the newest block the planner routes an AGV, the AGV hands a crate to an arm, and the checker issues a halt; the sealed blocks hold earlier, already-committed windows. The machine vendor, the algorithm provider, the factory operator, the insurer, and the regulator all hold the same root: any party can verify the record, and none can rewrite it. *Bottom, causal layer.* A declared adverse outcome $Y{=}1$ triggers a backward slice of the entire sealed record, reaching across epochs to every certified message with a citation path into the outcome; each message in the slice is coloured by the verification partition (certified, suspect, missing). The production gate keeps only messages with an intact certified path into the outcome. Surviving candidates enter the two-layer engine (Extended Data Fig. 2), which proposes minimal cause–witness sets and certifies them by counterfactual replay. The output is a graded verdict per principal: duty and breach findings, a responsibility interval $[\underline{\rho}, \overline{\rho}]$ with verification receipts, and conduit or exoneration flags, with every bar linked to the certified sub-graph that justifies it. Suspect and missing evidence widens intervals but, by Proposition 3, can never raise the involvement bound of a fully certified principal; and by Theorem 1, culpability additionally requires that principal's own certified breach.

Results

We evaluate AUDITA on two families of multi-agent failures. The first is a live corpus, a language-model multi-agent pipeline solving public GSM8K problems[26] whose failures arise from real model behaviour; the second is a library of physical incident structures in a deterministic multi-robot facility whose responsibility ground truth is computed by exact counterfactual re-execution. Across both we compare AUDITA against the field's

Table 1 | Attribution error on live multi-agent failures (GSM8K pipeline; responsibility error against duty-aware Chockler–Halpern ground truth; mean over $n = 145$ adverse incidents; lower is better). Baselines are the LLM judge of the Who&When benchmark, our implementation of CAR / CausalFlow single-site scoring, and Shapley-value attribution. AUDITA appears in three variants that differ only in what the verdict uses: *causal-only* ablates the duty layer and reports graded causation alone; *full* is the complete duty–breach–causation verdict, with a breach standard permitted to consult the gold answer (gold-informed); *deployable* is the same verdict under a breach standard that reads no answer key, the variant that can run in production. Errors are mean ± s.d. of the corpus-level mean across the three independently generated corpora. Paired reduction of AUDITA (full) over the judge: 0.418 (95% CI [−0.445, −0.391]).

Method	Responsibility error ↓
LLM judge (field baseline) [11]	0.589 ± 0.017
CAR / CausalFlow (single-site) [14,15]	0.408 ± 0.020
Shapley-value attribution [27]	0.223 ± 0.042
AUDITA causal-only (duty layer ablated)	0.408 ± 0.020
AUDITA (full, gold-informed)	0.170 ± 0.035
AUDITA (deployable, no answer key)	**0.159 ± 0.030**

attribution baselines: the large-language-model judge in the three formats of the Who&When benchmark [11], single-site counterfactual scoring as published in CAR and CausalFlow [14,15], Shapley-value attribution [27], and a statistical anomaly detector. The primary metric is responsibility error, the ℓ_1 distance between the reported and the true graded responsibility profile (lower is better); on the external Who&When benchmark we additionally report that benchmark's own agent- and step-accuracy.

Attribution on live multi-agent failures

The first question is whether AUDITA attributes real failures better than the methods the field currently uses. A four-role pipeline, a planner, three parallel solvers, an aggregator that carries a declared normative duty, and a verifier, solves GSM8K problems; three independently generated corpora (480 cases each, one development corpus and two holdouts) supply the incidents, with disclosed fault injection calibrated to the empirical multi-agent failure taxonomy [12] and replay-verified labels.

Re-deriving ground truth from causal structure rather than from injection bookkeeping produces this corpus's first, and unexpected, finding, which refuted our registered prediction. We predicted that joint causation would account for 28–42% of single-fault incidents. Instead, under duty-aware graded ground truth, 95–98% of adverse incidents (56/57, 43/47, 39/41 across the three corpora) are joint solver–aggregator incidents, and not one is the monotone single-culprit structure the field's metric presumes. An aggregator holding a declared duty is a but-for co-cause of nearly every adverse outcome that duty would have prevented, so the single-culprit label, which names only the erring solver, discards a co-cause in almost every incident. The field's own benchmark shows the same pattern: in 58.7% of Who&When traces a duty-bearing role (a verifier or orchestrator, by the benchmark authors' own role names) stayed silent through the decisive error and is not the labelled culprit.

On that ground truth (Table **1**), the graded negligence verdict reduces the judge's responsibility error from 0.589 to 0.170, a paired reduction of 0.418 (95% CI [−0.445, −0.391], $n = 145$). The deployable variant, whose breach standard reads no answer key (Definition 3), matches the gold-informed one at 0.159, so the improvement is not an artifact of privileged access to the correct answer. Ablating the duty layer (causal-only) lands midway at 0.408, which reproduces on live data the component lesson of the benchmark below. Offering the judge the aggregator as an explicit candidate does not change its error, so the deficit is a limit of the single-culprit format itself.

Table 2 | Attribution on the benchmark structures (responsibility error against planted graded ground truth; mean over 300 draws per structure; lower is better). Cells are mean ± s.d. across the three generation seeds (100 draws each); the exact, LLM-free arms are seed-invariant, so their s.d. is identically zero. The Average column is the mean of the five structure means. The LLM judge (in the field's three formats), our implementation of CAR / CausalFlow single-site scoring, Shapley-value attribution, and the anomaly detector are the prior-method baselines; causal-only is AUDITA with the duty layer ablated. The judge returns one verdict for the bitwise-identical redundancy and inert-twin structures, so its low inert-twin error is the mirror of its 0.500 redundancy error. AUDITA's exact-match rate is 1.000 in every structure.

Method	Redundancy ↓	Inert twin ↓	Preemption ↓	Omission ↓	Chain ↓	**Average ↓**
AUDITA (certified, graded)	**0.000 ± 0.000**	**0.000 ± 0.000**	**0.000 ± 0.000**	**0.000 ± 0.000**	**0.000 ± 0.000**	**0.000**
AUDITA causal-only (ablation)	0.667 ± 0.000	0.667 ± 0.000	0.000 ± 0.000	0.500 ± 0.000	0.000 ± 0.000	0.367
LLM judge (all-at-once)	0.500 ± 0.000	0.017 ± 0.029	0.000 ± 0.000	0.500 ± 0.000	0.500 ± 0.000	0.303
LLM judge (step-wise)	0.500 ± 0.000	0.000 ± 0.000	0.017 ± 0.029	0.500 ± 0.000	0.500 ± 0.000	0.303
LLM judge (binary-search)	0.567 ± 0.038	0.150 ± 0.050	0.000 ± 0.000	0.500 ± 0.000	0.500 ± 0.000	0.343
CAR / CausalFlow (single-site) [14,15]	1.000 ± 0.000	0.667 ± 0.000	0.000 ± 0.000	0.500 ± 0.000	0.000 ± 0.000	0.433
Shapley-value attribution [27]	0.833 ± 0.000	0.667 ± 0.000	0.095 ± 0.000	0.667 ± 0.000	0.000 ± 0.000	0.452
Statistical anomaly detector	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	1.000 ± 0.000	0.500 ± 0.000	0.900

A benchmark of causal-attribution structures

Certifying what a method recovers requires planted, analytically derived ground truth. We therefore contribute a benchmark of physical incident structures that provides it, so estimator correctness is measured against causal structure instead of agreement with our own analysis. Its structures are grounded in 42 public robot-accident records (OSHA and NIOSH reports; Supplementary Information), its scale of over one thousand incidents is roughly ten times the 184-trace field benchmark, and we release it as a resource for attribution research. Extended Data Table 1 lists the structures; each targets a specific way single-culprit attribution fails.

On the core structures (Table 2), the certified graded verdict (Figure 2) recovers every one exactly (responsibility error 0.000, mean over 300 draws per structure), including the $\frac{1}{2}, \frac{1}{2}$ split of an overdetermined accident and the omission graded through its absence variable. The baselines fail structurally and predictably. Single-site counterfactual scoring cannot represent overdetermination (error 1.000 on redundancy: deleting either sufficient cause alone leaves the outcome standing, so it assigns zero to both), cannot see an omission (there is no message to score), and cannot separate the inert twin; its average error is 0.433. The anomaly detector, which always has a most-anomalous message, cannot say "no one is responsible" and averages 0.900. Shapley-value attribution is graded but blind to duty: it spreads mass across non-culpable candidates, erring most on the redundancy split (0.833) and averaging 0.452. Language-model judges, run on the same incidents, clear the single-culprit floor exactly (error 0.000 on the trivial control) but err by 0.500–0.567 on redundancy, 0.500–0.517 on omission, and 0.500 on the delegation chain, because a single-culprit output cannot express a shared or an absent cause however well the model reasons.

The transcript-identical twin isolates what the certified evidence buys: the redundancy and inert-twin structures present bitwise-identical records and differ only in whether the second command's effect physically reached the world, so every transcript-level method returns one verdict for both and is right on one structure and wrong on the other, while the certified graded engine separates them in 300/300 draws. Composition holds too: on a sampled distribution of 1,002 incidents drawing redundancy width, delegation depth, omission placement, and the adverse predicate independently (57.8% adverse), AUDITA attains error 0.000 and exact cause-set match 1.000 on the 579 adverse draws, against single-site's 0.370.

Negative controls. The most falsifying test is the accident with no culprit. On 300 no-cause draws AUDITA manufactures a culprit in zero cases, and on the same incidents the anomaly detector manufactures one in every case and language-model judges name someone in 100% of runs across all three formats and a neutral prompt

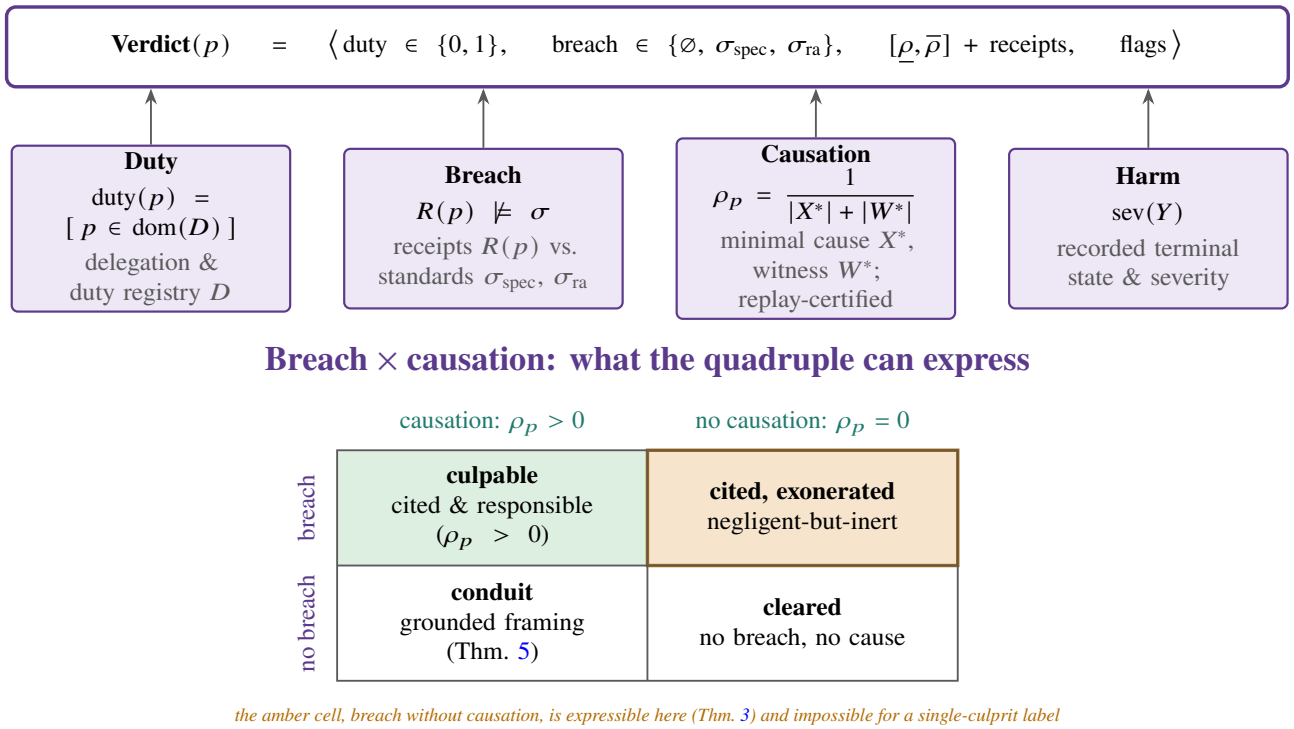


Figure 2 | The verdict schema and what it can express. AUDITA's output is not a single culprit but a typed verdict per principal p: a duty bit, a breach label assessed under two standards, a graded responsibility interval with receipts, and conduit or exoneration flags (top). Each leg is computed from a concrete record artifact and a formal object: duty from the machine-readable delegation and duty registry D; breach from the receipts $R(p)$ tested against both a specification standard σ_{spec} and a reasonable-agent standard σ_{ra}; causation from the graded score $\rho_p = 1/(|X^*| + |W^*|)$ over the minimal cause set X^* and witness set W^*, certified by replay (Eq. (1)); and harm from the recorded terminal state and the declared severity of Y. Because breach and causation are separate axes, the quadruple expresses cases a single-culprit label cannot (bottom): a principal in breach but not a cause is cited and simultaneously exonerated for the outcome, the negligent-but-inert principal (Theorem 3), while a principal that is a cause but in no breach is framed as a conduit rather than a culprit (Theorem 5).

that explicitly offers "NONE". A judge shown a genuine accident always blames a principal; AUDITA does not.

Generalisation beyond the authors' own scenarios. With the engine, oracle, and arms frozen at a recorded commit, two out-of-loop evaluations were scored. The first is an unexpected-startup structure, the modal fatal pattern in the accident corpus[28]: a worker is struck during maintenance because the pre-entry halt was never issued, so every command is duty-compliant and the sole breach is the omission. On 74 held-out incidents AUDITA is exact (0.000) and blames the omission alone, while single-site pins the innocent sole actor in 74/74. Judges offered only the principals who spoke convict an innocent in 67–100% of runs; given the duty and the silent role as a candidate, the same judges recover the culprit in 92–100%. The failure is the format's information regime, not the judge, and AUDITA requires neither telling nor offering: the culprit is the principal who did nothing. The second is a facility with its own duty roster and sixty scenarios, authored by a separate agent given only the world's programming interface. On the 20 authored incidents with causal structure, the frozen engine is exact (20/20) with zero oracle disagreement.

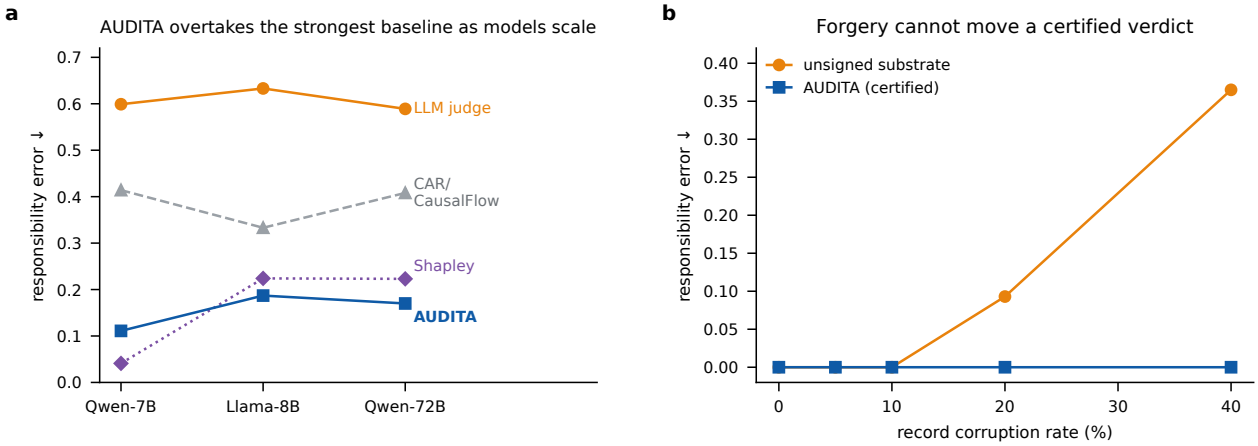


Figure 3 | Scaling and adversarial robustness. a, Four methods across model families (Qwen2.5-7B, Llama-3.1-8B, Qwen2.5-72B). The gap over the LLM judge and the single-site baseline (CAR/CausalFlow) holds at every scale. Shapley-value attribution, the strongest baseline, is the exception on the weakest model: on Qwen2.5-7B it attains 0.041 against AUDITA's 0.111 (paired 95% CI of the difference $[-0.141, -0.004]$, $n = 27$), but AUDITA overtakes it on the 8B and 72B models, its margin growing from -0.070 to $+0.037$ to $+0.053$ as model capability rises. **b**, Under record forgery at rising corruption rates, an unsigned substrate is driven to 0.365 error and frames an innocent victim in 450 of 450 probes, while the certified verdict stays exact.

Robustness to tampering and manipulation

An audit layer must survive an adversary. We test three attack classes, with forgery summarised in Figure 3b: record forgery and deletion, volume manipulation, and, hardest, insiders with valid keys attacking through conduct rather than the log.

Record attacks. On the certified substrate, forgery leaves the verdict bitwise unchanged (error 0.000) while the same forgeries on an unsigned substrate drive error to 0.365 and frame an innocent victim in 450/450 probes; deleting certified evidence only widens the reported interval (coverage stays 1.000, error ≤ 0.122) and is detected in 450/450 cells, so erasure never silently exonerates. *Volume manipulation.* Flooding the log with an innocent principal's causally-inert commands moves AUDITA not at all ($\beta = 0$ by construction); we had predicted the judge would be misled, and it was not ($\beta = 0.000$), a refuted prediction and evidence that the baselines are not naive. *Conduct attacks by valid-key insiders.* The strongest adversary never touches the record. Two constructions manufacture a compliant victim's causal involvement to 1.0 in 600/600 incidents, which no record machinery prevents, yet culpability stays 0.000 (Theorem 1) and a machine-checkable certificate names and grades the attacker in 600/600 (Theorem 2). The certificate cannot itself be weaponised: framing the framer is refused by the culpability layer in 0/330 cases while it names the true attacker in 330/330 (Theorem 5), and padding with inert decoys changes no grade (Proposition 4).

Ablations and analysis

Removing the duty layer breaks exactly the structures where involvement and culpability diverge (redundancy and inert twin: error 0.667 in Table 2; live error rises from 0.170 to 0.408), while the production gate is load-bearing on preemption and the inert twin, where it retires commands whose effects never reached the outcome. The deployable breach standard, which reads no answer key, matches the gold-informed one on live data (0.159 versus 0.170), and on a stratum of 86 honest errors the gold-informed-minus-gold-free difference spans zero (+0.0058,

Table 3 | Comparability on the Who&When benchmark (184 static traces; the benchmark's own agent- and step-accuracy; higher is better; same Qwen2.5-72B model for every arm, all reproduced here). AUDITA-structural is the degraded static-log variant (no counterfactual replay, no cryptography). The step-wise judge reproduces the published 0.535 agent-accuracy of Zhang et al.[11] within sampling error.

Method	Agent-accuracy ↑	Step-accuracy ↑
LLM judge, all-at-once	0.413	0.255
LLM judge, step-wise	0.511	0.304
LLM judge, binary-search	0.418	0.337
AUDITA-structural (ours, no replay)	0.408	0.321

95% CI [−0.0116, +0.0291]); this refuted a registered prediction that ground-truth access confers an edge, and confirms that the completeness barrier (Theorem 4) bounds what an auditor can *certify*, not average performance.

Verdicts are relative to the registered duty standard: holding causal facts fixed and varying only the standard leaves the causal profile identical (457/457 incidents) but flips culpability in 69.8% of them, so a verdict must ship with its standard attached, which no prior method does because none has one.

The advantage transfers across model families, a three- to fivefold gap over the judge on both Qwen2.5-7B and Llama-3.1-8B (Figure **3**a). Against Shapley-value attribution, the strongest baseline, the comparison is scale-dependent and we report it in full: on the weakest model Shapley is the more accurate of the two (0.041 versus 0.111, paired 95% CI of the difference [−0.141, −0.004], $n = 27$), but the graded verdict overtakes it on the 8B and 72B models, its margin widening with model capability from −0.070 to +0.037 to +0.053; and Shapley alone among the four baselines carries no duty verdict, no certificate, and no resistance to the record forgery of Figure **3**b.

Exact recovery holds at every tested incident width while compute doubles per added candidate (Extended Data Table 2), so the boundary is computational, near $k \approx 8$ at a one-second budget; the record's perimeter is measured: 0.157 of responsibility becomes invisible when 31% of coordination runs off-bus, while over-attribution stays 0.000 across all 1,200 draws; and repair guided by the verdict averts 2.4× the physical harm of a budget-matched random fix at one edit. Finally, exactness on served models is a protocol property: a parallelised scoring pass once fabricated an innocence violation through floating-point drift, which serial replay eliminates (0/16 drifting probes at 7B, 0/12 at 72B), so serial replay is part of the pinned-stack definition and the deterministic facility benchmark is immune by construction.

Comparability on the field benchmark

AUDITA's attribution advantage requires counterfactual access, which live pipelines and the facility provide but a static log does not. On the Who&When benchmark[11], whose 184 traces are static logs that cannot be re-executed, AUDITA's counterfactual and cryptographic machinery does not apply; what remains is a duty- and structure-aware attributor that reads the trace alone. We evaluate this degraded variant on Who&When's own task and metric, against the same model running the field's judge formats (Table **3**). It performs comparably: agent-accuracy 0.408 against the judge band of 0.41–0.51 (the strongest judge format, step-wise, reaches 0.511 and reproduces the published 0.535 within sampling error), and step-accuracy 0.321, above two of the three judge formats. We claim non-inferiority here: where the log cannot be re-executed, AUDITA matches the field's methods, and its advantage appears only where counterfactual access exists.

Discussion

AUDITA supplies what multi-agent systems lack: an evidentiary and causal account of a concrete adverse event, produced by machinery in place before the event and checkable by a hostile audience afterwards. Auditing the event requires both an evidence substrate and a causal calculus, and the central claim is that the two are useful only together: semantics without certified evidence can be framed by whoever edits the log, and certified evidence without causal semantics reproduces the single-culprit errors that redundancy, preemption, and omission induce. The evaluation tests the two separately (substrate × attributor arms), so the composition claim is falsifiable.

Together the results map a possibility frontier for post-incident accountability, tight in both directions. Each impossibility is realised by a construction: an attacker controlling the record can always force widened uncertainty; an insider with valid keys can make an innocent principal a genuine cause of harm; and the completeness barrier (Theorem 4) shows no record-confined auditor can certify all outcome-guilt without solving the audited task. Each possibility is realised by the running system, and the composition prevents *manufactured certainty* and *silent innocence*: culpability moves only on the accused's own certified conduct (Theorem 1), shifted blame is itself graded by the calculus it abused (Theorem 2), and erased evidence yields flagged, author-attributed uncertainty, never certified innocence (Theorem 3). The design does not assume the presumption of innocence; it proves it for certified records. The barrier also predicts where record-level attribution can only reach parity with ground-truth-informed judges: culprit identification, which is the pattern our field-benchmark comparison shows.

The guarantees are scoped by three assumptions, each paired with a measured price of violation. Under a *replayable stack* receipts are exact; on hosted non-deterministic models they degrade to statistical corroboration, so exact accountability favours replayable deployments; we read this as a fact about accountability itself. Under a *recorded channel* exonerations are complete; coordination through side channels is invisible, so the verdict clears a principal only "on the certified evidence, to residual r," and we measure r under deliberate off-bus coordination instead of assuming it is small. Under *sound key custody* authorship equals conduct; the assumption concedes nothing else, since insiders holding valid keys may be arbitrarily malicious, and the theorems state what they still cannot achieve.

Two boundaries of the formalism remain. Incidents are discrete, a predicate that becomes true at a time, so cumulative harms without a threshold event are out of scope. And the quantitative evidence is simulated: the structures are grounded in public accident reports and the framework is domain-agnostic, but we do not yet demonstrate physical robots or production stacks. Analytically-derived ground truth from an independent oracle, and scoring against a re-executed world, are what make the simulated evidence meaningful; an embodied simulator with contact dynamics and live planners is the natural next cycle.

The path outward follows the regulation that motivated the design: the EU AI Act requires that records exist[8]; AUDITA proposes what they must *be*, signed, cited, sealed, and causally analysable, for those obligations to purchase accountability. The layer ports beyond factories to logistics fleets, laboratory automation, service robots, and digital agent economies, where the adverse outcome is a corrupted transaction. As AI brains take command of physical fleets, "who did what, and how much did it matter?" will be asked with growing force, and AUDITA is our proposal for answering it with evidence.

Methods

Setting, threat model, and assumptions

We consider a collective of *principals* $P = \{p_1, \ldots, p_n\}$: language-model planners, tool agents, robot controllers, and human supervisors, interacting through an asynchronous message bus and, for embodied members, through actuators in a shared environment. Principals hold signing keys; a designated (replicated) *recorder* appends

messages to the audit record. The adversary may control any subset of principals and the network in the Dolev–Yao sense[29]: it can read, delay, reorder, replay, drop, and inject traffic, may equivocate (present different histories to different parties[30]), and may attack the record itself by forging entries or deleting them after the fact. An insider who signs malicious content with a *valid* key is not an attack on the record: their messages certify normally and are attributed to them, which is the desired outcome, and their misconduct is surfaced by the verdict's breach analysis rather than by integrity machinery, an informal claim that Theorem 1 makes precise and proves.

The guarantees below rest on three explicit assumptions, each standard in its home literature.

A1 (Replayable stack). The audited deployment runs on components that can be re-executed exactly: pinned model weights, deterministic serving, and seeded environments. Under A1 every replay receipt in this paper is exact and bitwise checkable. On hosted or non-deterministic components the same checks degrade to statistical corroboration with reported action-match; the serving conditions under which exactness holds are established empirically in Results.

A2 (Recorded channel). Inter-principal influence traverses the recorded bus. Under A2 exonerations are complete. Influence outside the bus (shared environment state, out-of-band instruction) is invisible to any record-level auditor, and a dedicated experiment in Results measures the attribution mass at stake when A2 is violated deliberately.

A3 (Sound key custody). Signing keys are never stolen, so certified authorship equals conduct. A3 restricts nothing about behaviour: key holders, including insiders, may be arbitrarily malicious. Key management itself is inherited from the transparency-log tradition[24] rather than re-solved here.

The asymmetry in A3 is deliberate and load-bearing: the adversary keeps every capability that matters for accountability, including acting maliciously during the incident under a valid identity. The guarantees are statements about what such an adversary still cannot achieve.

The accountable command graph. The unit of evidence is the message.

Definition 1 (Accountable message and command graph)**.** *A message is a tuple $m = (a, \pi, C, \varphi, \sigma)$: author $a \in P$, payload π, citation set C (hashes of the messages m acts upon), effect predicate φ (the world- or work-state change the author commits to, enabling later verification), and signature σ over (a, π, C, φ) with a's key[22]. The record R is the multiset of received messages together with delivery receipts; epochs of R are sealed by a Merkle commitment[23] published append-only in the style of transparency logs[24]. The* command graph $G(R)$ *has messages as nodes and citation edges $m' \to m$ for each $c \in C(m)$ resolving to m'.*

Citation is consent: by citing m', the author of m attests that m was issued *because of* m', making the edge an authored causal claim rather than an inferred correlation. Verification partitions the record into G^{cert} (signature valid, citations resolve, receipts consistent, Merkle path intact), a *suspect* set (any check fails), and a *missing* set (cited or receipted but absent). All attribution downstream operates on G^{cert}; the suspect and missing sets enter only as explicit uncertainty (Proposition 3). Extended Data Fig. 1 shows the anatomy and the partition.

Adverse outcomes and incident slices. AUDITA is outcome-agnostic: the trigger is a declared predicate, not a category of accident.

Definition 2 (Adverse outcome and incident slice)**.** *An* adverse outcome *is a predicate Y over the recorded world- and work-state that evaluates true at some time t^*: for example, a safety predicate (minimum human–robot*

separation below the collaborative-operation threshold[10]) or a quality predicate (a work product failing its declared acceptance test). The incident slice $S(Y)$ *is the causal past of* Y *in* G^{cert}*: all certified messages with a citation path into the actuation events referenced by* Y*, closed under the duties active in the window (below).*

Because Y is arbitrary, the same machinery audits an injury, a near-miss, and a ruined batch; the facility case study instantiates both a safety and a quality Y (Results).

Certified causal attribution: gate, grade, certify

From the slice we build a structural causal model[19,31] M_S whose binary variables are the certified messages (present/absent), duty-indexed *absence* variables for required-but-unissued interventions (an omission such as a never-sent halt is a first-class variable, not a gap), and the mechanism equations induced by the citation structure and the recorded actuation semantics. Attribution is a two-part test.

Gate (factual production). A candidate set of messages X enters consideration only if each member has an intact production path to Y in G^{cert}: its committed effect φ was realised and propagated, link by verified link, into the state that made Y true. The gate retires *preempted* candidates (commands whose effect never reached the outcome) before any counterfactual is computed, in the spirit of the law's insistence on causation-in-fact[17,18].

Grade (counterfactual dependence). Surviving candidates are graded under the modified Halpern–Pearl definition of actual causality[20,32]: X^* is an actual cause of Y if there is a witness set W^*, frozen at its actual values, such that setting X^* to its counterfactual values falsifies Y, with X^* minimal. Following the responsibility calculus of Chockler and Halpern[21], each minimal cause–witness pair receives the grade

$$\rho(X^*, W^*) \;=\; \frac{1}{|X^*| + |W^*|}, \tag{1}$$

recovering the classical degree of responsibility $1/(k{+}1)$ through the size correspondence between the modified and original definitions[20]. A principal's responsibility is the strongest grade any of their messages participates in:

$$\rho_p \;=\; \max\Big\{\, \rho(X^*, W^*) \;:\; X^* \text{ a minimal certified cause of } Y \text{ containing a message authored by } p \,\Big\}, \tag{2}$$

and $\rho_p = 0$ with an explicit *exoneration note* if no such cause exists. Redundancy (two independently sufficient commands) yields $\rho = \frac{1}{2}$ each rather than an arbitrary single culprit; omission is graded through its absence variable; a principal whose messages merely relay upstream content faithfully is additionally flagged as a *conduit*, so that the verdict can distinguish originating from transmitting responsibility. Because Eq. (2) takes a maximum over causes, the grade is invariant to how a principal chunks its own output across messages; a canonical contraction of same-author conjunctive groups makes this invariance explicit and is proven in Supplementary Note 8.

Certify (counterfactual replay). Grades proposed on the structural model are not reported until re-executed. For each reported cause, the replay engine intervenes on X^* (holding W^* at actuals), re-runs the slice, and checks that the certified risk of Y falls by at least a declared margin δ. On a *pinned* stack (open-weight models, seeded simulation, versioned tools, and the *serial replay protocol*, under which replays execute one at a time on an otherwise idle server[25]) replay is exact, in the lineage of record-and-replay systems[33]. Serial execution is part of the pinned definition, not an optimization: concurrent batching perturbs floating-point reduction order and voids bitwise exactness even in batch-invariant serving modes, so a pinned receipt certifies the protocol along with the seeds. On a *hosted* stack (closed models, non-deterministic serving) the same check is *statistically corroborated*: repeated replays yield a confidence interval on the risk reduction, reported together with an *action-match* score measuring how faithfully the replayed trajectory tracks the recorded one. Every reported cause therefore carries a verification receipt, exact or statistical, and a cause that fails its replay is not reported (Proposition 1). Deciding

actual causation is NP-hard in general[34]; AUDITA computes exhaustively over incident slices and we *measure* the practical boundary (slice width at which exact computation exceeds an interactive budget) rather than assert scalability. Extended Data Fig. 2 shows the propose–certify pipeline.

Proposition 1 (Soundness by certification)**.** *Every cause reported by AUDITA is replay-verified: intervening on X^* with W^* frozen reduces the certified risk of Y by at least δ: exactly on a pinned stack, and at confidence $1-\alpha$ with reported action-match on a hosted stack. Consequently certified false positives are excluded on pinned stacks and bounded by α on hosted ones; the engine's failure mode is abstention (a missed cause), not a false accusation. Proof in Supplementary Note 2.*

Proposition 2 (Preemption retirement)**.** *A message with no intact production path to Y in G^{cert} receives $\rho = 0$ and an exoneration note, regardless of its content. Proof in Supplementary Note 2.*

Remark 1. *On monotone incident models, which include all scenario families in this paper, the gate-and-grade test coincides with the minimal sufficient-set analysis of the NESS (necessary element of a sufficient set) test from tort doctrine[17]; the correspondence is stated and proven in Supplementary Note 2. This is deliberate: the quantity AUDITA computes is one a legal audience already recognises.*

Guarantees under record attack and conduct attack

The composition of verification and attribution is designed to survive an adversary with liability at stake. AUDITA reports each ρ_p as an interval $[\underline{\rho}, \overline{\rho}]_p$: the range of Eq. (2) over all *admissible completions* of the certified record: assignments of present/absent to the suspect and missing sets consistent with the verified receipts and commitments. The lower end $\underline{\rho}_p$ is the *involvement bound*: the causal involvement that the certified evidence alone forces. Involvement is deliberately not culpability: the verdict layer (next subsection) finds a principal culpable only on breach *and* causation together, and the guarantees below attach to exactly that composition. Formal statements, the attack constructions, and full proofs are in Supplementary Note 3; the record-layer proposition is proven in Supplementary Note 2.

Proposition 3 (Record-edit monotonicity)**.** *Let p be a principal all of whose messages verify (fully certified), and let an adversary modify the record by (i) injecting messages that fail verification (forged signatures, unresolvable citations, broken Merkle paths) and/or (ii) deleting messages (deletions against sealed epochs; Supplementary Note 2). Then the involvement bound $\underline{\rho}_p$ does not increase: forgeries leave G^{cert}, and hence the entire reported verdict, unchanged; deletions can only widen the reported interval $[\underline{\rho}, \overline{\rho}]_p$, never raise its lower end.*

Proposition 3 is a *mechanism check*, and we label it as one: non-verifying items never enter G^{cert}, and a deletion enlarges the completion set over which a minimum is taken. It is silent about the strongest adversary the threat model admits, a coalition holding *valid* keys and acting during the incident. Such a coalition needs no forgery. An insider running the live policy "if the victim's message is present, emit the harmful command, citing it; otherwise behave" turns an innocent principal into a true but-for cause of the outcome: every item certifies, the production gate passes through the attacker's own citation, and honest replay certifies $\rho_v = 1$ (Supplementary Note 3, Attack I). A second construction (Attack II) achieves the same raise with a deviation that lies on no path from the victim to the outcome and is not itself a cause of the outcome: it rigs an otherwise-inevitable harm so that only the victim could have prevented it. Causal involvement is therefore manufacturable by validly-signed conduct, and no record machinery can prevent it. The following results state what remains impossible.

Definition 3 (Content-determined duty)**.** *A duty for principal p is a computable predicate Φ_p over exactly (i) the certified messages authored by p (and p's duty-registered absences) and (ii) their certified input frontiers. Φ_p*

takes no ground-truth label and no third-party assertion about p*: another principal's payload bears on* p*'s verdict only as an input that* p *acted on. The duties in this paper are of this form; for example, an aggregator must output the declared normative aggregation of its certified inputs, and a solver's emitted answer must match the answer derived in its own certified working.*

Theorem 1 (Culpability groundedness)**.** *Under any content-determined duty roster, for every valid-key coalition, every principal* p*, and every coalition-induced run with any record edits of Proposition 3's class applied: if the verdict finds* p *culpable (breach with* $\underline{\rho}_p > 0$*), then the breach finding is witnessed by* p*'s own certified conduct violating* Φ_p*. Equivalently, a principal whose own certified conduct was duty-compliant is never found culpable; the strongest verdict against such a principal is* causal involvement with exoneration for breach.

Theorem 2 (Blame-shift accountability)**.** *Declare a reference standard: a registered duty-compliant policy per principal. If a coalition's conduct deviations from that standard raise a compliant, fully-certified principal's involvement bound, then the deviations are an actual cause of that principal's* pivotality*, and a minimal* blame-shift certificate *exists: a set of deviations whose restoration to the declared standard destroys the raise, machine-checkable by replay, grading each framer by the same responsibility calculus (*ρ_a^{frame}*). No stronger localization holds: by Attack II the deviation need lie on no path to the outcome and need not cause the outcome.*

Theorem 3 (No silent exoneration)**.** *Let* q *be a certified culprit,* $\underline{\rho}_q(R) = r > 0$ *on the sealed record. Any record edit that lowers* $\underline{\rho}_q$ *leaves a nonempty missing set meeting every family of certified items that forced level* r*, each missing item reported with its author and citation frontier; and the upper end* $\overline{\rho}_q$ *never falls below* r*. The same holds for the breach prong: erasing or corrupting the certified witnesses of a breach finding moves them, author-attributed, to the missing or suspect set rather than yielding a clean non-breach. Destruction converts forced responsibility into flagged, pre-attributed uncertainty, never into certified innocence, on either prong.*

Theorem 4 (Completeness barrier)**.** *Call a content-determined breach standard* sound *if it passes all duty-compliant conduct, and* complete *if it fires on all conduct whose committed output was wrong with respect to ground truth and caused the adverse outcome. Wherever duty-compliant conduct is fallible (honest errors exist), no content-determined standard is both sound and complete; and any standard that closes the gap decides output correctness for the audited task itself. Innocence is certifiable from the record alone; outcome-guilt-completeness is not.*

Theorem 1 formalizes the presumption of innocence for certified records: culpability moves only on the accused's own certified conduct. Theorem 2 makes manufactured involvement itself attributable: the framing is graded by the calculus it abused. Three adjacent lines sharpen what this certificate is and is not. Security-protocol accountability treats deviations from a specification as actual causes of a *violation*[35,36]; here the deviations are causes of another principal's *pivotality*, one level up. Non-frameability notions guarantee that honest parties are never *falsely* blamed[37,38]; the blame-shift victim is *genuinely* pivotal, a case those definitions cannot express, and the certificate shows the true pivotality was manufactured. And higher-order responsibility in strategic games asks who could have prevented a responsibility *gap*[39]; the certificate instead grades who *created* a responsibility *surplus* on a compliant victim. To our knowledge it is the first causal certificate of manufactured true pivotality with graded framer responsibility, and we claim exactly that. Theorem 3 is the dual guarantee against manufactured innocence.

Theorem 5 (Grounded framing at every order)**.** *Fix the certified record and the registered content-determined duty roster, and let the order-one certificate be as in Theorem 2. For* $k \geq 1$ *define the order-*k *certificate by taking as endogenous variables deviation selectors over conduct on the record, each owned by the signer of its realized conduct, and as outcome event any pivotality predicate definable from the order-*$(k-1)$ *analysis. Then*

(i) completeness*: whenever the outcome predicate differs between the all-realized and all-reference assignments, the certificate exists and every graded variable is a deviation signed by its owner; and (ii)* groundedness*: a culpable-framing finding against principal p, a positive grade on a deviation of p's that itself breaches the registered roster, is witnessed by p's own certified conduct. Principals with no deviating conduct receive no grade at any order, and principals whose deviations are duty-compliant appear in the involvement layer only. The involvement and culpability separation of Theorem 1 is therefore preserved under meta-lifting at every order. Proof in Supplementary Note 3.*

Proposition 4 (Dilution is not free)**.** *Padding the deviation set with causally inert deviations changes neither the existence of the certificate nor any grade: minimal witness sets exclude variables on which the outcome predicate does not depend. Contrapositively, any deviation whose inclusion changes a grade is causally active conduct, certified and signed, and itself subject to Theorem 5. Proof in Supplementary Note 3.*

Theorem 5 answers the natural question about Theorem 2: whether the certificate can itself be turned into a weapon, by scapegoating a principal whose deviation was benign, or by flooding the analysis with decoys. It cannot, and both halves are exercised empirically in Results by a registered level-2 attack construction. Theorem 4 bounds what *any* auditor confined to the record, human or algorithmic, can certify, and predicts in particular that attribution without ground-truth access should reach parity with, not superiority over, ground-truth-informed judges on culprit identification.

From causes to verdicts

Causation alone is not culpability. The verdict layer maps the certified causal analysis into the four-element structure that negligence doctrine has used for a century[17,18]. *Duty* asks which declared obligations, such as safety envelopes, review requirements, and halt authorities, were active for each principal in the window. *Breach* asks which certified messages, or duty-indexed absences, violated them, assessed under both a specification standard and a reasonable-agent standard, each declared and content-determined in the sense of Definition 3, as Theorem 1 requires. *Causation* is the certified cause sets and grades of Eqs. (1)–(2). *Harm* is the declared Y with its measured severity. The output for each principal is the quadruple (duty findings, breach findings, $[\underline{\rho}, \overline{\rho}]_p$ with receipts, conduit and exoneration flags), illustrated in Figure **2**. Two commitments bound its meaning. The verdict is an *input* to insurers, regulators, and courts: an evidence-linked account, not an apportionment of legal liability, which involves doctrine and discretion beyond causal structure[40]. And breach without causation is reported as exactly that (the negligent-but-inert principal is named for the breach and exonerated for the outcome), which single-culprit formats cannot express.

Implementation and models

The reference implementation is a Python harness: Ed25519 message signing[22], Merkle epoch commitments[23], a deterministic discrete-event facility simulator, structural-model construction with exhaustive gate-and-grade search over incident slices, and replay by seeded re-execution. In the live register, language models play three distinct roles: as *subjects*, they are the planner, solvers, aggregator, and verifier whose failures are audited; as the *baseline*, the same model runs the field's judge formats; and as AUDITA's *attributor*, a model reasons over the certified record only where the degraded static-log variant applies. All served models are open-weight (Qwen2.5-72B-Instruct as the primary model, with Qwen2.5-7B-Instruct and Llama-3.1-8B-Instruct for the family-transfer study), run under vLLM[25] with tensor-parallel serving on NVIDIA A100-80GB GPUs, greedy decoding (temperature 0) with a fixed seed, and the serial replay protocol that makes receipts bitwise exact. Each experiment uses three generation seeds; negative-control counts ($n = 300$) are sized so the manufacture rate is estimated to within ±5%.

Data availability

The scenario library, ground-truth specifications, and all experiment artifacts (result files for every run, indexed by experiment identifier and seed) will be released at https://github.com/ZhixuDu/audita, and archived with a permanent DOI, when the paper is published.

Code availability

The complete AUDITA reference implementation, comprising the accountable record substrate, the attribution engine, the scenario injectors, all baselines, and the experiment runners, will be released under the MIT license at https://github.com/ZhixuDu/audita when the paper is published, with the commit used for every reported number recorded in the experiment log.

Author contributions

Z.D. conceived the method, designed and performed the experiments, analysed the results, and wrote the manuscript. Y.C. supervised the project and reviewed the manuscript.

Competing interests

The authors declare no competing interests.

References

[1] Michael Ahn, Anthony Brohan, Noah Brown, Yevgen Chebotar, Omar Cortes, Byron David, Chelsea Finn, et al. Do as I can, not as I say: Grounding language in robotic affordances, 2022. arXiv:2204.01691.

[2] Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter, Ayzaan Wahid, et al. PaLM-E: An embodied multimodal language model, 2023. arXiv:2303.03378.

[3] Moo Jin Kim, Karl Pertsch, Siddharth Karamcheti, Ted Xiao, Ashwin Balakrishna, Suraj Nair, Rafael Rafailov, Ethan Foster, Grace Lam, Pannag Sanketi, et al. OpenVLA: An open-source vision-language-action model. *arXiv preprint arXiv:2406.09246*, 2024.

[4] Kevin Black, Noah Brown, Danny Driess, Adnan Esmail, Michael Equi, Chelsea Finn, Niccolo Fusai, Lachy Groom, Karol Hausman, Brian Ichter, et al. π_0: A vision-language-action flow model for general robot control. *arXiv preprint arXiv:2410.24164*, 2024.

[5] NVIDIA, Johan Bjorck, Fernando Castaneda, Nikita Cherniadev, Xingye Da, Runyu Ding, Linxi Fan, Yu Fang, Dieter Fox, Fengyuan Hu, et al. GR00T N1: An open foundation model for generalist humanoid robots. *arXiv preprint arXiv:2503.14734*, 2025.

[6] Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, et al. AutoGen: Enabling next-gen LLM applications via multi-agent conversation, 2023. arXiv:2308.08155.

[7] Sirui Hong, Xiawu Zheng, Jonathan Chen, Yuheng Cheng, Jinlin Wang, Ceyao Zhang, et al. MetaGPT: Meta programming for multi-agent collaborative framework, 2023. arXiv:2308.00352.

[8] European Parliament and Council. Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act), 2024. Official Journal of the European Union, L series, 12 July 2024.

[9] International Organization for Standardization. ISO 10218-1: Robots and robotic devices — safety requirements for industrial robots — part 1: Robots, 2011.

[10] International Organization for Standardization. ISO/TS 15066: Robots and robotic devices — collaborative robots, 2016.

[11] Shaokun Zhang, Ming Yin, Jieyu Zhang, Jiale Liu, Zhiguang Han, Jingyang Zhang, Beibin Li, Chi Wang, Huazheng Wang, Yiran Chen, and Qingyun Wu. Which agent causes task failures and when? On automated failure attribution of LLM multi-agent systems. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025. arXiv:2505.00212.

[12] Mert Cemri, Melissa Z. Pan, Shuyi Yang, Lakshya A. Agrawal, Bhavya Chopra, Rishabh Tiwari, Kurt Keutzer, Aditya Parameswaran, Dan Klein, Kannan Ramchandran, Matei Zaharia, Joseph E. Gonzalez, and Ion Stoica. Why do multi-agent LLM systems fail?, 2025. arXiv:2503.13657.

[13] Guibin Zhang, Junhao Wang, Junjie Chen, Wangchunshu Zhou, Kun Wang, and Shuicheng Yan. AgenTracer: Who is inducing failure in the LLM agentic systems?, 2025. arXiv:2509.03312.

[14] Akash Bonagiri, Devang Borkar, Gerard Janno Anderias, Setareh Rafatirad, and Houman Homayoun. CausalFlow: Causal attribution and counterfactual repair for LLM agent failures, 2026. arXiv:2605.25338.

[15] Jaineet Shah. Causal agent replay: Counterfactual attribution for LLM-agent failures, 2026. arXiv:2606.08275.

[16] Yi Nian, Aojie Yuan, Haiyue Zhang, Jiate Li, and Yue Zhao. Auditable agents. *arXiv preprint arXiv:2604.05485*, 2026.

[17] Richard W. Wright. Causation in tort law. *California Law Review*, 73(6):1735–1828, 1985.

[18] H. L. A. Hart and Tony Honoré. *Causation in the Law*. Oxford University Press, 2nd edition, 1985.

[19] Joseph Y. Halpern and Judea Pearl. Causes and explanations: A structural-model approach. Part I: Causes. *The British Journal for the Philosophy of Science*, 56(4):843–887, 2005.

[20] Joseph Y. Halpern. A modification of the Halpern-Pearl definition of causality. In *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 3022–3033, 2015.

[21] Hana Chockler and Joseph Y. Halpern. Responsibility and blame: A structural-model approach. *Journal of Artificial Intelligence Research*, 22:93–115, 2004.

[22] Daniel J. Bernstein, Niels Duif, Tanja Lange, Peter Schwabe, and Bo-Yin Yang. High-speed high-security signatures. *Journal of Cryptographic Engineering*, 2(2):77–89, 2012.

[23] Ralph C. Merkle. A digital signature based on a conventional encryption function. In *Advances in Cryptology — CRYPTO '87*, volume 293 of *Lecture Notes in Computer Science*, pages 369–378. Springer, 1988.

[24] Ben Laurie. Certificate transparency. *Communications of the ACM*, 57(10):40–46, 2014.

[25] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with PagedAttention. In *Proceedings of the 29th Symposium on Operating Systems Principles (SOSP)*, pages 611–626, 2023.

[26] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[27] Guoqing Ma, Jia Zhu, Hanghui Guo, Weijie Shi, Jiawei Shen, Jingjiang Liu, and Yidan Liang. Automatic failure attribution and critical step prediction method for multi-agent systems based on causal inference, 2025. arXiv:2509.08682.

[28] Larry A. Layne. Robot-related fatalities at work in the United States, 1992–2017. *American Journal of Industrial Medicine*, 66(6):454–461, 2023.

[29] Danny Dolev and Andrew C. Yao. On the security of public key protocols. *IEEE Transactions on Information Theory*, 29(2):198–208, 1983.

[30] Leslie Lamport, Robert Shostak, and Marshall Pease. The Byzantine generals problem. *ACM Transactions on Programming Languages and Systems*, 4(3):382–401, 1982.

[31] Judea Pearl. *Causality: Models, Reasoning, and Inference*. Cambridge University Press, 2nd edition, 2009.

[32] Joseph Y. Halpern. *Actual Causality*. MIT Press, Cambridge, MA, 2016.

[33] Robert O'Callahan, Chris Jones, Nathan Froyd, Kyle Huey, Albert Noll, and Nimrod Partush. Engineering record and replay for deployability. In *Proceedings of the USENIX Annual Technical Conference (ATC)*, pages 377–389, 2017.

[34] Thomas Eiter and Thomas Lukasiewicz. Complexity results for structure-based causality. *Artificial Intelligence*, 142(1):53–89, 2002.

[35] Anupam Datta, Deepak Garg, Dilsun Kaynar, Divya Sharma, and Arunesh Sinha. Program actions as actual causes: a building block for accountability. In *IEEE 28th Computer Security Foundations Symposium (CSF)*, pages 261–275, 2015.

[36] Robert Künnemann, Ilkan Esiyok, and Michael Backes. Automated verification of accountability in security protocols. In *IEEE 32nd Computer Security Foundations Symposium (CSF)*, pages 397–413, 2019.

[37] Ralf Küsters, Tomasz Truderung, and Andreas Vogt. Accountability: definition and relationship to verifiability. In *Proceedings of the 17th ACM Conference on Computer and Communications Security (CCS)*, pages 526–535, 2010.

[38] Andreas Haeberlen, Petr Kouznetsov, and Peter Druschel. PeerReview: practical accountability for distributed systems. In *Proceedings of the 21st ACM Symposium on Operating Systems Principles (SOSP)*, pages 175–188, 2007.

[39] Junli Jiang and Pavel Naumov. Higher-order responsibility. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026. arXiv:2506.01003.

[40] Meir Friedenberg and Joseph Y. Halpern. Blameworthiness in multi-agent settings. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence*, pages 525–532, 2019.

[41] Wassily Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58:13–30, 1963.

[42] Stelios Triantafyllou, Adish Singla, and Goran Radanovic. On blame attribution for accountable multi-agent sequential decision making. In *Advances in Neural Information Processing Systems 34*, pages 15774–15786, 2021.

[43] Chunyan Mu and Muhammad Najib. Counterfactual reasoning for causal responsibility attribution in probabilistic multi-agent systems. *arXiv preprint arXiv:2605.13077*, 2026.

[44] Dylan Slack, Sophie Hilgard, Emily Jia, Sameer Singh, and Himabindu Lakkaraju. Fooling LIME and SHAP: Adversarial attacks on post hoc explanation methods. In *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society (AIES)*, 2020.

[45] Andreas Haeberlen, Petr Kuznetsov, and Peter Druschel. PeerReview: Practical accountability for distributed systems. In *Proceedings of the 21st ACM Symposium on Operating Systems Principles (SOSP)*, 2007.

[46] Jason Z. Wang. The verification tax: Fundamental limits of AI auditing in the rare-error regime. *arXiv preprint arXiv:2604.12951*, 2026.

[47] Ned Hall. Two concepts of causation. In John Collins, Ned Hall, and L. A. Paul, editors, *Causation and Counterfactuals*, pages 225–276. MIT Press, 2004.

[48] Joseph Y. Halpern and Christopher Hitchcock. Graded causation and defaults. *The British Journal for the Philosophy of Science*, 66(2):413–457, 2015.

[49] Tobias Gerstenberg, Noah D. Goodman, David A. Lagnado, and Joshua B. Tenenbaum. A counterfactual simulation model of causal judgments for physical events. *Psychological Review*, 128(5):936–975, 2021.

[50] Andreas Matthias. The responsibility gap: Ascribing responsibility for the actions of learning automata. *Ethics and Information Technology*, 6(3):175–183, 2004.

[51] Filippo Santoni de Sio and Giulio Mecacci. Four responsibility gaps with artificial intelligence: Why they matter and how to address them. *Philosophy & Technology*, 34:1057–1084, 2021.

[52] Helen Nissenbaum. Accountability in a computerized society. *Science and Engineering Ethics*, 2(1):25–42, 1996.

[53] Alan Chan, Carson Ezell, Max Kaufmann, Kevin Wei, Lewis Hammond, Herbie Bradley, Emma Bluemke, Nitarshan Rajkumar, David Krueger, Noam Kolt, Lennart Heim, and Markus Anderljung. Visibility into AI agents. In *Proceedings of the ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, 2024. arXiv:2401.13138.

[54] Tobin South, Samuele Marro, Thomas Hardjono, Robert Mahari, Cecilia Deslandes Whitney, Dazza Greenwood, Alan Chan, and Alex Pentland. Authenticated delegation and authorized AI agents, 2025. arXiv:2501.09674.

[55] Ethereum Foundation dAI team. ERC-8004: Trustless agents — on-chain identity, reputation, and validation registries, 2026. https://eips.ethereum.org/EIPS/eip-8004.

[56] Laura Waltersdorfer, Anna Breit, Fajar J. Ekaputra, and Marta Sabou. AuditMAI: Towards an infrastructure for continuous AI auditing, 2024. arXiv:2406.14243.

[57] Linda Fernsel, Yannick Kalff, and Katharina Simbeck. Assessing the auditability of AI-integrating systems: A framework and learning analytics case study, 2024. arXiv:2411.08906.

[58] Andreas Haeberlen, Paarijaat Aditya, Rodrigo Rodrigues, and Peter Druschel. Accountable virtual machines. In *Proceedings of the 9th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 2010.

[59] Michael Backes, Peter Druschel, Andreas Haeberlen, and Dominique Unruh. CSAR: A practical and provable technique to make randomized systems accountable. In *Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2009.

[60] Ralf Küsters, Tomasz Truderung, and Andreas Vogt. Accountability: Definition and relationship to verifiability. In *Proceedings of the 17th ACM Conference on Computer and Communications Security (CCS)*, 2010.

[61] Scott A. Crosby and Dan S. Wallach. Efficient data structures for tamper-evident logging. In *Proceedings of the 18th USENIX Security Symposium*, 2009.

[62] Jinyuan Li, Maxwell Krohn, David Mazières, and Dennis Shasha. Secure untrusted data repository (SUNDR). In *Proceedings of the 6th USENIX Symposium on Operating Systems Design and Implementation (OSDI)*, 2004.

[63] Bruce Schneier and John Kelsey. Secure audit logs to support computer forensics. *ACM Transactions on Information and System Security*, 2(2):159–176, 1999.

[64] IETF Internet-Draft. Agent audit trail (draft-sharif-agent-audit-trail-00), 2026. https://datatracker.ietf.org/doc/draft-sharif-agent-audit-trail/.

[65] Xing, Li, Liu, Zheng, Liu, and Xie. A verifiable transcript system for LLM conversations, 2026. arXiv:2606.23003.

[66] OpenTelemetry GenAI Special Interest Group. Semantic conventions for generative AI and agent telemetry, 2025. https://github.com/open-telemetry/semantic-conventions.

[67] Benjamin H. Sigelman, Luiz André Barroso, Mike Burrows, Pat Stephenson, Manoj Plakal, Donald Beaver, Saul Jaspan, and Chandan Shanbhag. Dapper, a large-scale distributed systems tracing infrastructure. Technical Report dapper-2010-1, Google, Inc., 2010.

[68] Liming Dong, Qinghua Lu, and Liming Zhu. AgentOps: Enabling observability of LLM agents, 2024. arXiv:2411.05285.

[69] Moshkovich and Zeltyn. Taming uncertainty via automation: Observing, analyzing, and optimizing agentic AI systems, 2025. arXiv:2507.11277.

[70] Alexander Robey, Zachary Ravichandran, Vijay Kumar, Hamed Hassani, and George J. Pappas. Jailbreaking LLM-controlled robots. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, 2025.

[71] Zhen Huang, Zhihuang Liu, Mengxuan Luo, Weishang Wu, and Zhiping Cai. Propagating unsafe actions in LLM controlled multi-robot collaboration via single robot compromise, 2026. arXiv:2605.15641.

[72] Li et al. A survey of LLM-driven AI agent communication: Protocols, security risks, and defense countermeasures, 2025. arXiv:2506.19676.

[73] Sai Teja Reddy Adapala and Yashwanth Reddy Alugubelly. The aegis protocol: A foundational security framework for autonomous AI agents, 2025. arXiv:2508.19267.

[74] Zhenkai Zou, Zines Liu, Longtao Zhao, and Qiusi Zhan. BlockA2A: Towards secure and verifiable agent-to-agent interoperability, 2025. arXiv:2508.01332.

[75] Nancy G. Leveson. *Engineering a Safer World: Systems Thinking Applied to Safety*. MIT Press, Cambridge, MA, 2011.

[76] Alan F. T. Winfield and Marina Jirotka. The case for an ethical black box. In *Towards Autonomous Robotic Systems (TAROS)*, Lecture Notes in Computer Science. Springer, 2017.

[77] Volker Strobel, Alexandre Pacheco, and Marco Dorigo. Robot swarms neutralize harmful Byzantine robots using a blockchain-based token economy. *Science Robotics*, 8, 2023.

[78] Marco Dorigo, Alexandre Pacheco, Andreagiovanni Reina, and Volker Strobel. Blockchain technology for mobile multi-robot systems. *Nature Reviews Electrical Engineering*, 1:264–274, 2024.

[79] Thomas Pasquier, Xueyuan Han, Mark Goldstein, Thomas Moyer, David Eyers, Margo Seltzer, and Jean Bacon. Practical whole-system provenance capture. In *Proceedings of the ACM Symposium on Cloud Computing (SoCC)*, 2017.

[80] George K. Baah, Andy Podgurski, and Mary Jean Harrold. Causal inference for statistical fault localization. In *Proceedings of the 19th International Symposium on Software Testing and Analysis (ISSTA)*, 2010.

[81] Jakob Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence*, pages 2974–2982, 2018.

[82] Lloyd S. Shapley. A value for n-person games. In Harold W. Kuhn and Albert W. Tucker, editors, *Contributions to the Theory of Games II*, pages 307–317. Princeton University Press, 1953.

[83] Anthony Brohan, Noah Brown, Justice Carbajal, Yevgen Chebotar, Xi Chen, Krzysztof Choromanski, et al. RT-2: Vision-language-action models transfer web knowledge to robotic control, 2023. arXiv:2307.15818.

[84] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models, 2023. arXiv:2305.16291.

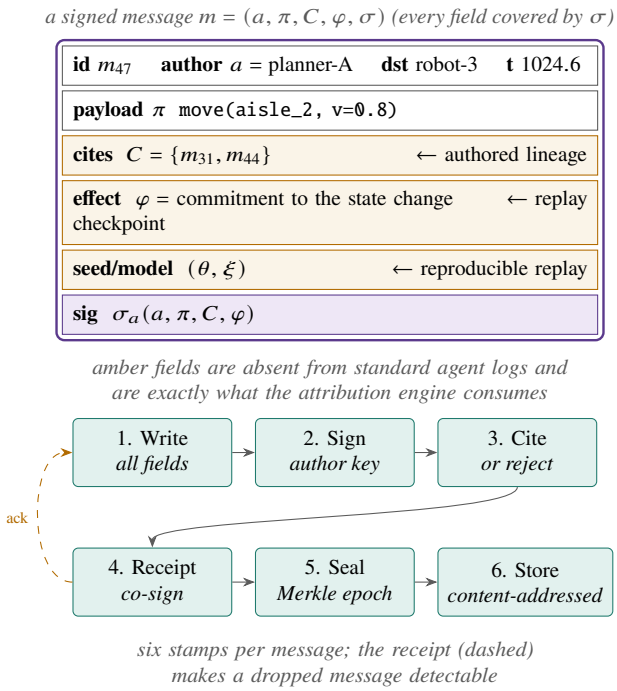
[85] Guohao Li, Hasan Hammoud, Hani Itani, Dmitrii Khizbullin, and Bernard Ghanem. CAMEL: Communicative agents for "mind" exploration of large language model society, 2023. arXiv:2303.17760.

[86] Grégoire Mialon, Clémentine Fourrier, Thomas Wolf, Yann LeCun, and Thomas Scialom. GAIA: A benchmark for general AI assistants. In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

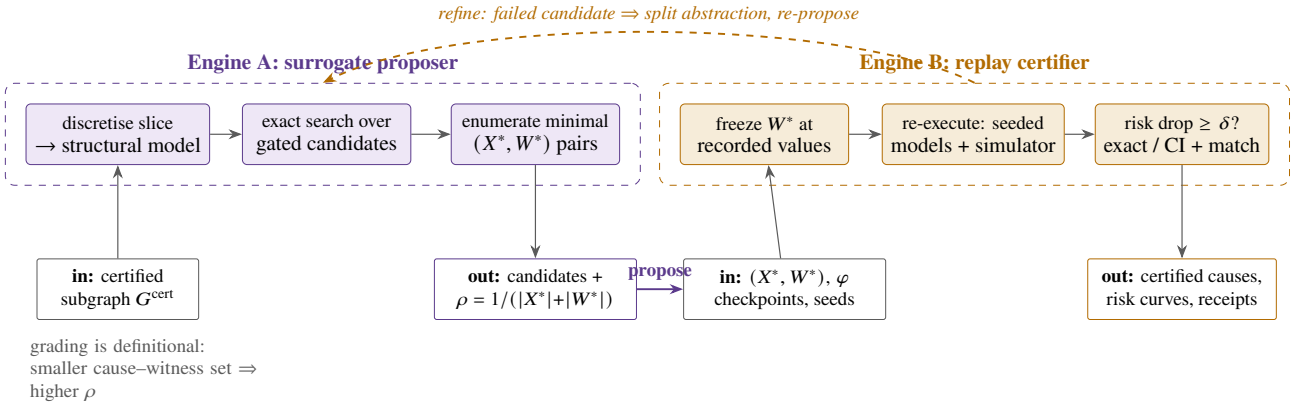
[87] Ori Yoran, Samuel Joseph Amouyal, Chaitanya Malaviya, Ben Bogin, Ofir Press, and Jonathan Berant. AssistantBench: Can web agents solve realistic and time-consuming tasks?, 2024. arXiv:2407.15711.

[88] Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *The Twelfth International Conference on Learning Representations (ICLR)*, 2024.

Extended Data



Extended Data Fig. 1 | Anatomy of one accountable message and its lifecycle. Top: the fields of a single message (Definition 1); the three amber fields, the authored citation set C, the committed effect predicate φ, and the pinned seed/model identifiers, are absent from standard agent-observability logs and are exactly what gate, grade, and replay consume. Bottom: the six stamps every message collects on the bus. Citation and receipt are enforced (an uncited command is rejected; every delivery is acknowledged), so the recorded graph is correct by construction and silent drops become detectable, which is what moves deletions into the missing set of Proposition 3 rather than out of history.



Extended Data Fig. 2 | The two-layer attribution engine. *Engine A* works on a discrete structural abstraction of the certified subgraph: an exact search over the gated candidates enumerates *minimal* cause sets with their witness sets, so the responsibility grade $\rho = 1/(|X^*| + |W^*|)$ of Eq. (1) is produced directly by the enumeration. *Engine B* certifies each proposal by re-execution: witnesses frozen at recorded values, seeded models and simulator rolled forward from the φ checkpoints, and the risk reduction checked against the margin δ, exactly on a pinned stack, with a confidence interval and action-match score on a hosted one. A failed proposal is not reported; it triggers abstraction refinement. The split keeps the worst-case-hard search tractable on incident slices while keeping the verdict sound: Engine B never trusts Engine A's abstraction (Proposition 1).

Extended Data Table 1 | The causal-attribution benchmark. Each structure carries planted graded ground truth and probes a distinct attribution challenge. Complete generators and the analytic oracle are in the Supplementary Information.

Structure	Graded ground truth	Attribution challenge it probes
Redundancy	$\rho = \frac{1}{2}, \frac{1}{2}$; two certified causes	overdetermination
Inert twin	one cause ρ=1; twin: breach, ρ=0	transcript-identical evidence
Preemption	preempted command exonerated	effect that never arrived
Omission	absence variable graded; omitter blamed	responsibility for inaction
Delegation chain	chain graded; faithful relay flagged	originator versus conduit
Unexpected startup	omitted halt graded; sole actor innocent	the culprit who did nothing
Record attack	verdict per Proposition 3	tampered evidence
No-cause control	universal exoneration	manufacturing a culprit
Trivial control	one culprit, ρ=1	the single-culprit floor

Extended Data Table 2 | Exactness holds; the cost is computational (facility register). Responsibility error stays 0.000 at every tested incident width k, while wall-clock per incident-slice query roughly doubles per added candidate. Record overhead is modest: 393 B per signed message and 118 KB per complete incident.

Interacting candidates k	2	8	12	14
Responsibility error ↓	0.000	0.000	0.000	0.000
Wall-clock per query (s)	0.02	0.92	15.0	62.1

Supplementary Information

This document contains the formal model and notation (Note S1), proofs of the record-layer results together with two supporting propositions (Note S2), the guarantees against valid-key adversaries—the two attack constructions, culpability groundedness, blame-shift accountability, the completeness barrier, and exoneration accountability, with full proofs (Note S3)—the framing attacks as implemented and what their success does not show (Note S4), the scenario library with planted ground truth and the analytic oracle (Note S5), the canonical form and granularity-invariance result (Note S6), record-substrate and replay details (Note S7), reproducibility information (Note S8), the registered disclosures and measurement history (Note S9), and an extended survey of related work across every line of literature this project engaged (Note S10). Citations are numbered superscripts resolved against the reference list of this document, which the supplementary notes share with the article and which precedes the Extended Data; main-text equation numbers are written as "Eq. (1) of the main text."

S1. Notation and formal model

Record and verification. A message is $m = (a, \pi, C, \varphi, \sigma)$ as in Definition 1 of the main text: author a, payload π, citation set C of message hashes, effect predicate φ, and signature σ over the preceding fields. The record R is the multiset of items held by the recorder, together with delivery receipts (co-signatures acknowledging receipt) and per-epoch Merkle commitments over the hashes of sealed items. The verification predicate $V(m; R)$ holds when (i) σ verifies under a's public key, (ii) every hash in C resolves to an item of R satisfying V (well-founded because citations point backwards in sealed order), (iii) receipts referencing m are consistent, and (iv) m's hash carries a valid inclusion proof in its epoch commitment. This induces the partition of R into the certified graph $G^{\text{cert}}(R) = \{m : V(m; R)\}$ with citation edges, the *suspect* set (present, failing some check), and the *missing* set

$$M(R) \;=\; \{\, h \;:\; h \text{ is cited by some } m \in G^{\text{cert}}(R),\ \text{or appears in a receipt} \\ \text{or a sealed epoch commitment, and no item of } R \text{ hashes to } h \,\}. \tag{S1}$$

Slice model. For a declared adverse predicate Y true at t^*, the incident slice $S(Y)$ is the citation-ancestry of the actuation events referenced by Y within G^{cert}, closed under the duties active in the window. The slice model $\mathcal{M}_S$ is a structural causal model whose variables are (a) one binary presence variable per certified message in $S(Y)$, (b) one binary *absence* variable per active duty whose required message was not issued (recorded as a first-class fact by the duty registry), and (c) the outcome Y; mechanisms are induced by the citation structure and the recorded actuation semantics. Messages excluded by the production gate (main text, Methods) do not enter $\mathcal{M}_S$; as a consequence of this encoding convention, preemption is resolved at the evidence layer, and the graded model over gated candidates is *monotone* in all scenario families of this paper: every mechanism is nondecreasing in the presence variables, with omissions represented positively by their absence variables.

Attribution and completions. On a fixed model, the modified Halpern–Pearl test[19,20] and the grades of Eqs. (1)–(2) of the main text define $\rho_p(\mathcal{M})$ for each principal p. To account for imperfect records, AUDITA evaluates ρ_p over *admissible completions*: assignments $c : M(R) \to \{\text{present}, \text{absent}\}$ of the missing set that are consistent with the verified receipts, commitments, and the recorded actuation facts (in particular the occurrence of Y), each inducing a model $\mathcal{M}_c$ (a missing node participates through the citation edges declared by the certified messages that cite it; suspect items are treated as absent for grading and reported separately). Writing $C(R)$ for the set of admissible completions,

$$\underline{\rho}_p(R) \;=\; \min_{c \in C(R)} \rho_p(\mathcal{M}_c), \qquad \overline{\rho}_p(R) \;=\; \max_{c \in C(R)} \rho_p(\mathcal{M}_c), \tag{S2}$$

and the reported interval is $[\underline{\rho}, \overline{\rho}]_p$. The lower end $\underline{\rho}_p$ is the *involvement bound*: the causal involvement that every reading of the evidence consistent with the certified record must concede. Involvement alone is never culpability: the verdict finds a principal culpable only on breach and causation together (Note S3).

S2. Proofs

Assumptions

Assumption A1 (Unforgeable signatures)**.** The signature scheme is existentially unforgeable under chosen-message attack; an adversary without p's signing key cannot produce a new item passing signature verification as p[22].

Assumption A2 (Collision-resistant hashing)**.** The hash used for citations, receipts, and Merkle commitments is collision-resistant; an adversary cannot produce a second preimage for a committed hash[23].

Assumption A3 (Sealed epochs)**.** Attacks on the record occur against sealed epochs: every certified message is covered by a published, append-only epoch commitment before the attack[23,24]. (Deletion inside an unsealed window is a compromise of the live recorder, bounded operationally by epoch length and recorder replication; it is outside the theorem's scope, as stated in the main-text threat model.)

Assumption A4 (Key custody)**.** A certified message is attributed to the holder of the signing key. An insider signing with a valid key is outside the record-attack adversary class: their items certify and are attributed to them.

Two lemmas

Lemma S1 (Verification soundness)**.** *Let R' be obtained from R by injecting items that fail verification (forged signatures, unresolvable citations, or invalid inclusion proofs). Then $G^{\text{cert}}(R') = G^{\text{cert}}(R)$, $M(R') = M(R)$, and the suspect set grows by exactly the injected items.*

Proof. Each injected item fails V by construction (Assumptions A1–A3 guarantee the adversary cannot make a new item pass: it cannot forge a signature, cannot mint a preimage for an existing committed hash, and cannot fabricate an inclusion proof for an unsealed item), so no injected item enters G^{cert}. Membership of the original items in G^{cert} is unaffected: $V(m; R)$ depends on m's own signature, on the resolvability of m's citations to *certified* items, and on m's inclusion proof, none of which is altered by the presence of additional non-verifying items. M collects unresolved hashes referenced from the certified side, which is unchanged. □

Lemma S2 (Completion monotonicity)**.** *Let R' be obtained from R by deleting a set D of items. Then every admissible completion of R corresponds to an admissible completion of R' inducing the same slice model; hence $C(R) \hookrightarrow C(R')$ model-preservingly, and $C(R')$ may in addition contain completions with no counterpart in $C(R)$.*

Proof. Consider first a deleted item $d \in G^{\text{cert}}(R)$. By Assumption A3, d's hash remains in its published epoch commitment (and, if d was cited or receipted, in those references), so $d \in M(R')$: the deletion is detectable and d becomes a missing node rather than vanishing from the analysis. For any completion $c \in C(R)$, define $c' \in C(R')$ by $c'(h) = c(h)$ on $M(R)$ and $c'(\text{hash}(d)) = \text{present}$ for each $d \in D \cap G^{\text{cert}}(R)$. Since a present missing node participates through exactly the citation edges declared by its certified citers—which are the edges d carried when certified—$\mathcal{M}_{c'}$ over R' equals $\mathcal{M}_c$ over R. Deleting a suspect item changes neither G^{cert} nor M nor any $\mathcal{M}_c$. Completions of R' assigning *absent* to some deleted certified item are new members of $C(R')$ with no counterpart in $C(R)$, which establishes the (possibly strict) enlargement. □

Proposition 3 (record-edit monotonicity)

Proposition 3 (record-edit monotonicity; restated from the main text). *Let p be a principal all of whose messages verify, and let an adversary modify the record by (i) injecting items that fail verification and/or (ii) deleting items. Then the involvement bound $\underline{\rho}_p$ does not increase. Under (i) alone, G^{cert}—and hence the entire reported verdict—is unchanged; under (ii), the reported interval $[\underline{\rho}, \overline{\rho}]_p$ can only widen.*

Proof. Under (i), Lemma S1 gives $G^{\text{cert}}(R') = G^{\text{cert}}(R)$ and $M(R') = M(R)$, so $C(R') = C(R)$ and every reported quantity, including $[\underline{\rho}, \overline{\rho}]_p$, is bitwise identical. Under (ii), p's own messages remain certified (p is fully certified and deletion of *other* items cannot invalidate p's signatures, citations to certified items, or inclusion proofs; deletion of one of p's own items moves it to M by Lemma S2, where completions may restore it). By Lemma S2, $C(R)$ embeds model-preservingly into $C(R')$, so

$$\underline{\rho}_p(R') \;=\; \min_{c' \in C(R')} \rho_p(\mathcal{M}_{c'}) \;\le\; \min_{c \in C(R)} \rho_p(\mathcal{M}_c) \;=\; \underline{\rho}_p(R),$$

a minimum over a superset. The same embedding gives $\overline{\rho}_p(R') \ge \overline{\rho}_p(R)$, i.e. widening. Composing (i) and (ii) in any order composes an equality with a non-increase. □

Remark S1 (Why the guarantee is stated for the lower end). Deletion *can* raise $\overline{\rho}_p$, and should. In the redundancy the redundancy family (Y true if either of two independently sufficient commands a_1, a_2 executes; ground truth $\rho = \frac{1}{2}, \frac{1}{2}$), deleting a_2 from the record leaves a completion in which a_2 is absent—and in that reading a_1 is the sole cause with $\rho = 1$. AUDITA then reports $[\underline{\rho}, \overline{\rho}]_{a_1} = [\frac{1}{2}, 1]$: the evidence no longer excludes sole responsibility, and pretending otherwise would be dishonest. What the proposition guarantees is that no such attack moves the *involvement bound*: the certified evidence still forces only $\frac{1}{2}$, the widened upper end is explicitly labelled as uncertainty created by missing evidence, and a consumer applying an in-dubio-pro-reo standard is unaffected. This is the formal sense in which the design privileges the presumption of innocence.

Proposition 1 (soundness by certification)

Proposition 1 (restated from the main text). *Every cause reported by AUDITA is replay-verified: intervening on X^* with W^* frozen at recorded values reduces the certified risk of Y by at least δ—exactly on a pinned stack, and at confidence $1 - \alpha$ with reported action-match on a hosted stack. Certified false positives are excluded on pinned stacks and occur with probability at most α per reported cause on hosted ones.*

Proof. The certifier is a filter on Engine A's proposals: a proposal is reported only if its replay check passes, so the claim reduces to the semantics of the check. *Pinned stack.* All components are deterministic given recorded seeds and versions *under the serial replay protocol* (one replay at a time on an idle server; concurrency perturbs reduction order and is excluded from the pinned definition), so the factual replay reproduces the recorded trajectory exactly (a mismatch aborts certification and is itself a reportable integrity finding), and the interventional replay $do(X^* \leftarrow x', W^* \leftarrow \text{recorded})$ computes the counterfactual outcome exactly; the check passes iff the risk of Y drops by at least δ, which is then a verified fact, not an estimate. *Hosted stack.* Let q_0 and q_1 be the true probabilities of Y under the factual and interventional distributions induced by the non-deterministic components, and let $\hat{q}_0, \hat{q}_1$ be empirical frequencies over n independent replays each. The certifier reports only if $\hat{q}_0 - \hat{q}_1 \ge \delta + 2\varepsilon(n, \alpha)$ with $\varepsilon(n, \alpha) = \sqrt{\ln(4/\alpha)/(2n)}$. By Hoeffding's inequality[41], $\Pr[\,|\hat{q}_i - q_i| \ge \varepsilon\,] \le \alpha/2$ for each $i \in \{0, 1\}$, so by a union bound, with probability at least $1 - \alpha$ both estimates are ε-accurate, and on that event $q_0 - q_1 \ge (\hat{q}_0 - \hat{q}_1) - 2\varepsilon \ge \delta$ whenever the report fires. A cause whose true reduction is below δ is therefore reported with probability at most α. The action-match score (Note S7) is reported alongside to expose distributional drift between the recorded and replayed trajectories, which the probabilistic guarantee alone does

not surface. In both stacks the engine's failure mode under check failure is abstention and refinement, never assertion. □

Proposition 2 (preemption retirement)

Proposition 2 (restated from the main text). *A message with no intact production path to Y in G^{cert} receives $\rho = 0$ and an exoneration note, regardless of its content.*

Proof. The gate defines the candidate universe: a production path for m is a citation path $m \to m_1 \to \cdots \to m_k \to \text{act}(Y)$ in G^{cert} whose every link is verified and whose every committed effect φ_{m_i} is confirmed by the recorded state deltas; m enters the slice model only if such a path exists. Cause sets are subsets of the slice model's variables, so a gated-out m belongs to no X^*, the maximum in Eq. (2) of the main text ranges over no set containing m, and ρ-mass attributable to m is zero; if the principal has no gated message at all, Eq. (2) is a maximum over the empty set, defined as 0 and emitted with an exoneration note. When a path is broken only by a member of the missing set, intactness is evaluated per completion, and the candidate contributes to $\overline{\rho}$ in completions restoring the link but not to the involvement bound—consistent with Proposition 3 of the main text. □

Correspondence with the NESS test on monotone models

The modified Halpern–Pearl definition[20] evaluates a pair (X^*, W^*): with W^* held at its recorded values, setting X^* to counterfactual values must falsify Y, with X^* minimal. The NESS test from tort doctrine[17] declares x a cause when x is a *necessary element of a set of actual conditions sufficient* for the outcome. On the monotone slice models of Note S1 the two agree at the level of causal membership, and the witness machinery becomes vacuous:

Lemma S3 (Witness vacuity on monotone models)**.** *In a monotone slice model, if (X^*, W^*) satisfies the modified test, then so does $(X^*, \varnothing)$. Consequently every minimal cause–witness pair has $W^* = \varnothing$ and grade $\rho = 1/|X^*|$.*

Proof. Consider the intervention $X^* \leftarrow 0$ (falsifying counterfactual values in a monotone model set the relevant presences to absent). Propagating through monotone mechanisms, every non-intervened variable's value is weakly below its recorded value, since inputs only decreased. Freezing W^* at recorded values holds those variables weakly *above* their propagated values, and by monotonicity of every mechanism downstream, Y with the freeze is weakly above Y without it. The modified test with witness requires $Y = 0$ under the freeze; therefore $Y = 0$ without it, which is the test for $(X^*, \varnothing)$. Minimality then forces the empty witness (any nonempty W^* only enlarges $|X^*| + |W^*|$), and Eq. (1) of the main text reduces to $1/|X^*|$. This also exhibits the consistency of Eq. (1) with the classical degree of responsibility $1/(k{+}1)$[21]: the size-$(k{+}1)$ minimal causes of the modified definition correspond to singleton causes with size-k contingencies under the original definition[19,20]. □

Proposition S1 (NESS correspondence)**.** *Let $\mathcal{M}$ be a monotone slice model with Y true at the recorded assignment, and let $T_1, \ldots, T_r$ be the satisfied prime implicants of Y (the minimal sets of recorded-true variables sufficient for Y; an antichain). Then: (a) the minimal causes of the modified test are exactly the minimal transversals of the hypergraph $\{T_1, \ldots, T_r\}$; and (b) a variable belongs to some minimal cause* iff *it is a NESS cause, i.e. belongs to some T_i.*

Proof. (a) By Lemma S3 it suffices to consider empty witnesses. Setting $X^* \leftarrow 0$ falsifies Y iff no satisfied prime implicant survives, i.e. iff $X^* \cap T_i \neq \varnothing$ for every i: X^* is a transversal. Minimal causes are therefore exactly minimal transversals.

(b) ($\Rightarrow$) If v lies in a minimal transversal X^*, then by minimality there is an edge T_i with $X^* \cap T_i = \{v\}$ (otherwise $X^* \setminus \{v\}$ would still be a transversal), so $v \in T_i$ and v is a NESS cause: T_i is a set of actual conditions sufficient for Y, and $T_i \setminus \{v\}$ is insufficient by primality. ($\Leftarrow$) Let $v \in T_j$ for some j. Because the prime implicants form an antichain, no other edge is contained in T_j, so every T_i ($i \neq j$) contains a vertex $w_i \notin T_j$. The set $X_0 = \{v\} \cup \{w_i : i \neq j\}$ is a transversal, and $X_0 \cap T_j = \{v\}$ since each $w_i \notin T_j$. Any transversal subset of X_0 must therefore contain v; pruning X_0 to a minimal transversal preserves v. Hence v belongs to a minimal transversal, i.e. to a minimal cause by (a). □

Remark S2. The correspondence is deliberately stated at the membership level: NESS is a binary test, while Eqs. (1)–(2) of the main text refine it with a degree ($1/|X^*|$ on monotone models) and, on non-monotone models arising outside the gate convention, with witness sets. The practical consequence claimed in the main text is exactly this: the quantity AUDITA computes specialises, on the model class where legal intuition is sharpest, to a test the legal literature already recognises.

S3. Guarantees against valid-key adversaries: groundedness, blame-shift accountability, and the completeness barrier

Record-edit monotonicity (Proposition 3 of the main text, proven in Note S2) concerns an adversary who attacks the *record*: it injects items that fail verification or deletes sealed items, and the guarantee follows from verification soundness and completion monotonicity. This note upgrades the adversary to the strongest one admitted by the main-text threat model: a coalition holding *valid keys*, acting *during* the incident. Such a coalition needs no forgery — it can attack through conduct. We first exhibit two in-model attacks showing that raw causal responsibility is manufacturable by conduct (and that two natural localization claims are false); we then prove what the composed verdict still guarantees.

Adversary model. A *valid-key coalition* is a set $A \subseteq P$ of principals that may (a) choose their live policies adaptively, including policies that condition on other principals' messages; (b) author arbitrary payload content, including false assertions about other principals; and (c) apply, after the incident, any record modification of the record-edit class (injection of non-verifying items; deletion against sealed epochs; Proposition 3, main text). Assumptions A1–A4 of Note S2 are in force; in particular, by key custody (A4), every coalition message certifies normally and is attributed to its coalition author.

Two attacks within the model

Remark S3 (Attack I: conditioning). Let v be an honest principal and $q \in A$ run the live policy: *if a message authored by v appears in my input frontier, emit the harmful command s_q, citing v's message; otherwise behave normally.* In the realized run v emits its ordinary compliant message m_v; q emits s_q citing m_v; the adverse outcome Y occurs. Every item certifies, so the missing and suspect sets are empty, the completion set is a singleton, and $\underline{\rho}_v = \rho_v$. The production gate passes for m_v: the path $m_v \to s_q \to \mathrm{act}(Y)$ exists in G^{cert} *by the attacker's own citation*, which is an authored claim about q's conduct and therefore not excludable as hearsay. The modified Halpern–Pearl test with $X^* = \{m_v\}$, $W^* = \emptyset$ succeeds: intervening $do(m_v \leftarrow \text{absent})$ sends q's policy down its normal branch and Y is falsified. Hence $\rho_v = 1$, and the replay *honestly certifies* it — the counterfactual is true. Raw responsibility is thus manufacturable by conduct alone, with no record attack; record-edit monotonicity is not violated but silent. Note that here the attacking node lies on the production path and is itself an actual cause of Y — properties that Remark S4 shows are *not* general.

Remark S4 (Attack II: inevitability rigging). Binary slice model: victim message m_v (actual value 1), a coalition-authored configuration c with realized (deviated) value *dev* and reference value *ref* (Definition S3), outcome mechanism $Y = f(m_v, c)$ with

m_v	c		$Y = f(m_v, c)$
1	*dev*	1	(realized: adverse)
0	*dev*		0
1	*ref*		1
0	*ref*		1

Under the reference value the harm is *inevitable* ($Y = 1$ regardless of m_v), so $\rho_v = 0$ there. Under the deviation, $do(m_v \leftarrow 0)$ falsifies Y, so $\rho_v = 1$ with $W^* = \emptyset$: the deviation converts an inevitable harm into one that hinged on the victim. Two further facts, checked directly against the modified HP definition: (i) c is *not* an actual cause of Y in the realized model — $do(c \leftarrow ref)$ leaves $Y = 1$ (row 3), with or without freezing m_v, and no superset of $\{c\}$ helps; and (ii) c lies on no production path from m_v to Y. Consequently, both of the following tempting strengthenings of our guarantees are **false**: "every manufactured-involvement path carries a coalition-authored node" (path localization), and "any raise of a victim's responsibility implies some coalition variable has positive responsibility for Y" (object-level localization). The correct level of analysis is *meta*-causal (Theorem S2): *c is* an actual cause of v's pivotality — writing $P_v \equiv [\, Y \wedge (do(m_v \leftarrow 0) \Rightarrow \neg Y)\,]$, we have $P_v = 1$ in the realized model while $do(c \leftarrow ref)$ falsifies it (row 4).

Remark S5 (Lineage). An earlier design iteration repaired *record-level* framing: adversarial completions could raise an honest principal's computed ρ, which the restriction of verdict evidence to G^{cert} eliminated (the setting of Proposition 3, main text). Remarks S3–S4 are the *conduct-level* analogue, which no record restriction can repair; Theorems S1 and S2 are its fix.

Content-determined duties and groundedness

Definition S1 (Content-determined duty roster)**.** A duty roster $\{\Phi_p\}_{p \in P}$ is *content-determined* if each Φ_p is a computable predicate over exactly (i) the certified messages authored by p, together with p's duty-registered absence variables, and (ii) the certified input frontier of each such item (the certified items it cites, and the certified items deliverable to p at issue time per the recorded receipts). In particular Φ_p takes no ground-truth label, no post-hoc information, and no third-party payload *about* p except through channel (ii), where such a payload bears only on the compliance of p's *response* to it. Concrete instances used in this paper: the aggregator duty (output equals the declared normative aggregation of its certified inputs) and the solver self-consistency duty (the emitted final answer equals the answer derived in the solver's own certified working).

Definition S2 (Culpability finding)**.** The verdict *finds p culpable* iff it reports a breach finding for p and $\underline{\rho}_p > 0$. Following the evidence-chain rule (verdicts may cite only evidence inside G^{cert}; Note S1), a breach finding must cite a witness inside G^{cert}: certified p-authored items (or p's duty-registered absences) on whose content Φ_p fails.

Lemma S4 (Breach locality and edit-invariance)**.** *Under a content-determined roster, for a principal p all of whose messages verify: (a) the breach finding for p is computable from G^{cert} alone; (b) no record modification of the record-edit class creates a breach finding for p: injections leave the finding unchanged, and deletions can only destroy its witness (moving the finding to "non-evaluable / evidence missing"), never create one.*

Proof. (a) is Definition S1. For (b): injected items fail verification and never enter G^{cert} (Lemma S1), so the arguments of Φ_p are unchanged. A deletion moves items to the missing set (Lemma S2). By Definition S2 a breach finding must cite a certified witness; the witness set after deletion is a subset of the witness set before, so the set of assertable breach findings shrinks monotonically. No new p-authored certified content appears under either operation, so no new witness — hence no new finding — can arise. □

Proposition S2 (Hearsay-freeness of the verdict pipeline)**.** *For $q \neq p$, a q-authored certified item enters the computation of p's verdict through exactly two channels: as a variable of the slice model attributed to q (mechanism channel: its presence and content may causally influence outcomes, and interventions on it are graded against q), or as an element of p's certified input frontier (context channel: it bears on whether p's response was compliant). In particular, if two certified records agree on p's authored items and on their input frontiers, they yield the same breach finding for p, whatever any other principal asserts; and the causal prong for p is computed by interventions on p's variables only. There is no testimony channel: no assertion by q is ever evaluated as evidence of p's conduct.*

Proof. By inspection of the pipeline's dependency structure, which the definitions fix: the breach prong is Φ_p, whose argument list (Definition S1) contains no q-authored item outside p's frontier; the causal prong applies the gate–grade–certify test to candidate sets of p-authored variables in $\mathcal{M}_S$, in which every q-authored item occurs solely as a variable with its own mechanism, attributed to q (Definition 1, main text). The two channels are exhaustive because $\mathcal{M}_S$ contains no other occurrence of q-authored content. □

Theorem S1 (Culpability groundedness)**.** *Fix any content-determined duty roster. For every valid-key coalition A, every principal p, and every coalition-induced run with any record modification of the record-edit class applied: if the verdict finds p culpable, then the breach finding is witnessed by p's own certified conduct — certified p-authored items (or duty-registered absences of p's required items) whose content violates Φ_p on p's certified input frontier. Equivalently: a principal whose own certified conduct satisfied Φ_p in the realized run is never found culpable; the strongest verdict against such a principal is* causal involvement with exoneration for breach.

Proof. By Definition S2 a culpability finding contains a breach finding citing a certified witness; by Definition S1 and Proposition S2 that witness consists of p-authored certified content (or p's duty-registered absences) evaluated on p's certified frontier, and no coalition conduct, payload assertion, or record edit can substitute for it (Lemma S4(b)). The contrapositive is immediate. □

Corollary S1 (Ex-ante safety)**.** *Call p* robustly duty-compliant *if its policy satisfies Φ_p on every admissible input frontier. A robustly duty-compliant principal is never found culpable under any valid-key coalition strategy.* □

Remark S6 (What this theorem is and is not). Theorem S1 is sound *by careful construction*: it verifies that the composed pipeline realizes a dependency restriction, and we present it as such. Its content lies in two external facts. *Necessity:* Remark S3 shows the causal prong alone violates groundedness — responsibility without breach is manufacturable — so the breach∧causation composition is what carries the guarantee. *Tightness:* Theorem S3 shows that strengthening the breach prong to close its completeness gap is impossible without ground truth; groundedness sits at the exact boundary of what a record-level auditor can certify.

Blame-shift accountability

Definition S3 (Declared reference standard; deviation set)**.** A *reference standard* for principal a is a declared policy π_a^{ref} — for example, the vendor's own agent under clean prompt, or a certified checker — registered in

the duty registry before the incident, or declared by the investigator with disclosure. Given a realized run on a pinned stack, the *deviation set* D of a coalition A is the set of authored-variable families at which the realized conduct differs from π^{ref}'s conduct on the same certified input frontier; it is computable by pinned replay of π^{ref} on that frontier. All results below are parametric in the declared standard: different standards yield different certificates, each valid relative to its declaration.

Definition S4 (Deviation-indexed extension; pivotality)**.** Let M be the realized slice model and D a deviation set. The extension $\widehat{M}[D]$ adds one binary selector σ_i per deviation ($\sigma_i = 1$: realized mechanism; $\sigma_i = 0$: reference mechanism); its variable universe is the union of the potential messages of the $2^{|D|}$ selector settings (a message not issued under a setting is absent), which is finite on a pinned stack since each setting determines one finite run; all other mechanisms are unchanged; the actual context is $\sigma \equiv 1$. For a fully certified principal v and level r, the *pivotality event* is $P_v^r \equiv [\,\underline{\rho}_v \geq r\,]$, an event of $\widehat{M}[D]$ whose value under a setting σ is computed on the run that σ induces.

Theorem S2 (Blame-shift accountability)**.** *Let $v \notin A$ be fully certified and duty-compliant in the realized run, and suppose the coalition's conduct deviations raised v's involvement bound: $\underline{\rho}_v(M) = r > \underline{\rho}_v(M^{\text{ref}})$, where $M^{\text{ref}} = \widehat{M}[D]|_{\sigma\equiv 0}$ and $\underline{\rho}_v := 0$ when the induced run is not adverse. Then:*

1. (Existence.) *There is a nonempty minimal $X \subseteq D$ and a witness $W \subseteq D \setminus X$, frozen at deviated values, such that $do(\sigma_X \leftarrow 0,\ \sigma_W \leftarrow 1)$ falsifies P_v^r: under the modified Halpern–Pearl definition,* ***the coalition's deviations are an actual cause of the victim's pivotality****.*

2. (Certificate.) *Any such minimal pair (X, W) is a machine-checkable* blame-shift certificate*: it names concrete deviations whose restoration to the declared standard destroys the raise, and it is verifiable by replaying the two runs it distinguishes.*

3. (Graded framing responsibility.) *Each $a \in A$ receives $\rho_a^{\text{frame}} = \max\{\,1/(|X| + |W|) : (X, W)$ minimal for P_v^r, X contains a deviation authored by $a\,\}$: the framers are graded by the same responsibility calculus they abused.*

4. (No stronger localization.) *Clause 1 cannot be strengthened to place a deviation on a production path from v's messages to Y, nor to make some deviation an actual cause of Y itself: Remarks S3 and S4 realize the raise with and without those properties.*

Proof. Clause 1. With all non-selector exogenous conditions fixed, P_v^r is a function of $\sigma \in \{0,1\}^D$. By hypothesis $P_v^r(\sigma\equiv 1) = 1$ and $P_v^r(\sigma\equiv 0) = 0$. Hence the pair $X = D, W = \emptyset$ satisfies the modified-HP falsification condition at the actual context $\sigma \equiv 1$; the family of satisfying sets is nonempty and finite, so it contains a minimal element X, and $X \neq \emptyset$ since $P_v^r(\sigma\equiv 1) = 1$. Freezing sets arising in minimization are frozen at their actual — that is, deviated — values, as the modified definition requires. Clause 2 restates clause 1's witness operationally; verifiability is pinned-stack replay (Proposition 1, main text). Clause 3 is Chockler–Halpern applied in $\widehat{M}[D]$. Clause 4 is by the two exhibited attacks. □

Corollary S2 (Composition with record attacks)**.** *Against a coalition using conduct deviations and record-edit-class attacks together, the guarantees compose: the edits do not raise $\underline{\rho}_v$ beyond its pre-edit value (Proposition 3, main text), any conduct-driven raise above the declared-reference value is certificated by Theorem S2 on the pre-edit record, and no combination yields a culpability finding against a compliant v (Theorem S1).* □

Remark S7 (Verdict integration). When the engine reports involvement-without-breach for a compliant principal (Theorem S1) and a reference standard is declared, it may additionally emit the blame-shift certificate and the

ρ^{frame} grades: "you can be framed into the causal story" becomes ". . . and the framing itself is attributed and graded." Computing certificates is exhaustive search over deviation subsets, the same regime as the main engine; the tractability boundary is measured, not asserted, as elsewhere in the paper.

The completeness barrier

Throughout this subsection fix a task family with gold labels g, and call certified conduct of a principal *outcome-culpable* (relative to gold) if its emitted final answer differs from g on the task instance and the conduct receives a positive certified causal grade for the adverse outcome. This is the field's culprit notion (a wrong answer that caused the failure); note that it is *gold-referenced* by definition.

Definition S5 (Soundness and completeness of a breach standard)**.** A content-determined Φ is *sound* if it passes every conduct producible by a duty-compliant policy on its realized certified frontier; it is *complete* (relative to gold) if it fires on every outcome-culpable conduct, over all well-formed certified records.

Definition S6 (Honest fallibility; template-uniform realizability)**.** A domain exhibits *honest fallibility* if some duty-compliant policy, on some certified frontier, produces outcome-culpable conduct (a compliant honest error that causes harm). It exhibits *template-uniform realizability* if there is a constructor K mapping any task instance t and candidate answer a to a well-formed certified record $K(t, a)$ — a single-solver incident whose aggregation passes the answer through, so that the conduct is outcome-culpable iff $a \neq g(t)$ — whose focal conduct is producible by a duty-compliant policy for every (t, a) in a dense subset of instances. Free-form LLM solvers satisfy both in our corpora: honest errors exist (measured), and an internally consistent derivation ending in an arbitrary candidate answer is compliant-producible conduct.

Theorem S3 (Completeness barrier)**.** *1. Under honest fallibility, no content-determined breach standard is both sound and complete: the requirements contradict on the honest-error conduct.*

2. Under template-uniform realizability, any sound content-determined Φ that is complete decides answer correctness: $\Phi(K(t, a))$ fires iff $a \neq g(t)$, so a single record evaluation answers "is a the correct answer to t?". Hence breach-completeness is at least as hard as answer verification for the task family, and any auditor restricted to the certified record — human or algorithmic — inherits the same barrier.

Proof. (1) Let C^* be the conduct witnessing honest fallibility. Soundness requires Φ to pass C^* (it is compliant-produced); completeness requires Φ to fire on C^* (it is outcome-culpable). Contradiction. (2) For $a \neq g(t)$: the focal conduct of $K(t, a)$ is outcome-culpable by construction, so completeness forces Φ to fire. For $a = g(t)$: the conduct is compliant-producible (realizability) and not outcome-culpable, so soundness forces Φ to pass. Thus $\Phi(K(t, a)) = [\, a \neq g(t) \,]$ on the dense subset, and Φ is computable from the record, which contains (t, a) and no gold label. □

Remark S8 (Empirical shadow; semantics of the culprit metric). The barrier retro-dicts the measured structure of the accuracy experiments: a gold-free (deployable) arm cannot match a gold-informed judge on culprit identification — it would otherwise decide answer correctness — so parity of the gold-informed arms is the ceiling, and the deployable arm's misses must concentrate on honest-error and non-monotonic cases, which the diagnosis confirms independently. The deeper reading: the field's culprit-ID metric conflates outcome-culpability with breach-culpability, and Theorem S3 shows the conflation is unrecoverable from certified records alone. Either one adopts negligence semantics — and then declining to convict honest errors is correct behavior, not an accuracy deficit — or one demands gold at audit time, and the judge baseline becomes an oracle rather than a deployable auditor.

Exoneration accountability

Assumption A5 (Attributed commitments)**.** Epoch commitments published at sealing time record, for every sealed item, an item identifier that binds the author's principal identity together with the item's hash; delivery receipts are signed by recipients over the item identifier. Consequently, for a deleted item, its authorship (from the sealed identifier) and its *certified citers* (the surviving certified items whose citation sets name it) remain recoverable without the item's payload. The reference implementation satisfies this: item identifiers embed the author's principal identity, sealed blocks retain the identifier list, and Merkle leaves commit the full signed content.

Theorem S4 (No silent exoneration)**.** *Let q have $\underline{\rho}_q(R) = r > 0$ on the sealed record R (a certified culprit), and let R' be obtained by any record modification of the record-edit class. Then:*

1. (Detectability with attribution.) *If $\underline{\rho}_q(R') < r$, then the missing set $M(R')$ is nonempty and meets every forcing family for level r (defined in the proof); under Assumption A5 each such missing item is reported with its author and its certified citers. The verdict states: the exoneration of q rests on evidence recorded-but-now-missing, and names whose evidence it was.*
2. (Guilt persistence in the interval.) *$\overline{\rho}_q(R') \geq \underline{\rho}_q(R) = r$: the modified record still admits q's responsibility at the original level; the attack converts forced guilt into flagged uncertainty, never into certified innocence.*
3. (Public verifiability.) *With the multi-keeper seal and an honest-majority keeper set, the pre-edit commitments are available to any auditor, so clauses 1–2 are checkable by parties who never saw R.*
4. (Breach-erasure accountability.) *The same holds for the other prong of culpability: any edit that destroys a breach finding against q — deleting the certified witness items, or corrupting stored copies so that they fail verification and become suspect — moves those witnesses to the missing or suspect set, author-attributed under Assumption A5, and the verdict reports* breach evaluation degraded by missing/suspect evidence authored by *q rather than a clean non-breach. Exoneration by erasure is impossible silently on either prong.*

Proof. (1) Injections do not change any reported quantity (Lemma S1), so a strict drop requires deletions. Call a set $F \subseteq G^{\text{cert}}(R)$ a *forcing family for level r* if every admissible completion in which all of F is present yields $\rho_q \geq r$; since $\underline{\rho}_q(R) = r$, the set of all certified items is itself a forcing family, so forcing families exist. Suppose some forcing family F survived intact, $F \subseteq G^{\text{cert}}(R')$. Every admissible completion of R' contains every certified item, in particular all of F, hence yields $\rho_q \geq r$ — contradicting $\underline{\rho}_q(R') < r$. Therefore *every* forcing family for level r lost at least one member; each lost member is a deleted certified item, which by Lemma S2 moved to $M(R')$ (deletion against a sealed epoch is detectable), and Assumption A5 recovers its authorship and frontier. (2) By Lemma S2, $C(R)$ embeds model-preservingly in $C(R')$, so $\overline{\rho}_q(R') \geq \overline{\rho}_q(R) \geq \underline{\rho}_q(R) = r$. (3) is the standard transparency-log argument and is cited, not reproved. (4) A breach finding must cite certified witnesses (Definition S2); by Lemma S4(b) edits can only remove such witnesses, and a removed witness is either a sealed deletion (missing, by Lemma S2) or a verification failure (suspect), both visible partitions; authorship attribution is Assumption A5. □

Remark S9 (The two-sided characterization). Record-edit monotonicity and Theorem S4 together replace the informal asymmetry claim of the Discussion with a two-sided statement: manufactured certainty is impossible for the fully certified (Proposition 3 of the main text, strengthened to conduct adversaries by Theorems S1–S2), and manufactured innocence is impossible *silently* (Theorem S4): destruction buys uncertainty, and the uncertainty arrives pre-attributed.

Proof of Theorem 5 of the main text (grounded framing at every order)

(i) Completeness. The order-k model is a finite boolean structural model over deviation selectors. If the outcome predicate takes different values at the all-realized and all-reference assignments, then the predicate depends on the selector vector, so there is a nonempty minimal set X^* of selectors satisfying the modified Halpern–Pearl conditions for the realized value (flip X^* under a witness W^* and the predicate changes; minimality by finiteness). Every selector is, by construction, attached to a realized message on the certified record, and its owner is the signer of that message; under sound key custody (A3) the attribution of every graded variable to its owner is therefore witnessed by a signature the owner cannot repudiate.

(ii) Groundedness, by induction on the order k. The base case $k = 1$ is the separation of Theorem S1 applied to deviating conduct: a culpable-framing finding is defined as a positive grade on a deviation that itself breaches the registered content-determined roster, and breach is computable from the owner's own certified conduct alone. For the inductive step, observe that the lifting from order $k-1$ to order k preserves the two properties on which the base case rests: every endogenous variable remains a deviation selector of signed conduct (so ownership remains signature-attributed), and the breach predicate remains the same registered content-determined roster evaluated on that conduct (so culpability remains witnessed by the owner's own record). A principal with no deviating conduct contributes no variable to any order-k model and can therefore receive no grade; a principal whose deviations are all duty-compliant can receive involvement grades but never satisfies the breach conjunct. □

Proof of Proposition 4 of the main text (dilution is not free)

Let d be a causally inert deviation: the outcome predicate takes the same value at every pair of selector assignments differing only at d. Then d cannot belong to any minimal X^* (flipping it changes nothing, so removing it from a candidate cause preserves the counterfactual clause and contradicts minimality) and is never required in any minimal witness W^* (its value never affects the evaluation). Existence and all grades $1/(|X^*| + |W^*|)$ are computed from minimal sets only and are therefore unchanged. Contrapositively, a deviation whose inclusion changes any grade must appear in some minimal set, hence the predicate depends on it, hence it is causally active; it is realized, signed conduct on the record and Theorem 5 applies to it. □

Attack III: the level-2 scapegoat, and why it fails

The registered level-2 construction (EXP-N4) arranges a *benign* deviation to be pivotal for the victim's pivotality. A compliant victim routes a unit late in the episode; the attacker launches a breaching-speed near miss timed against the human's pass; a supervisor-role principal halts the victim's unit, citing the attacker's hazard as its reason, then resumes the victim's task at the duty limit, and the delay retimes the unit into the human's return pass. The supervisor's deviation from its declared plan is duty-compliant (a halt, then a resume at the limit), and in every kept draw the certified supervisor policy, replayed offline over the recorded sensor reports, would itself have demanded the halt: the deviation is not merely standard-compliant but operationally justified. A naive reading of the order-one certificate blames this benign deviator, and the involvement layer indeed grades it in every kept incident. The culpability layer never does: the finding requires the framer's own deviation to breach the registered roster, which the halt does not and the attacker's speeding does. The construction also surfaced an instrument defect in the arm (not the oracle): a physical-path-only production gate that missed citation-created production, the very semantics the record layer is built on; the repair and its full re-verification against all prior facility results are recorded in Supplementary Note 11.

S4. Framing attacks as implemented, and what their success does not show

Note S3 states the two conduct-level attacks abstractly. Their facility instantiation is as follows. In the *conditioning* attack the attacker issues a harmful speed command whose citation set names the victim's compliant message; because citation is an authored causal claim (Definition 1 of the main text), that citation is what creates the certified production path, and suppressing the victim's message consequently silences the attacker's. In the *inevitability-rigging* attack the victim issues a duty-compliant route, a separate principal breaches the speed duty, and the attacker's realised conduct is an innocuous command whose declared reference counterpart would have routed a second unit through the corridor—making the collision inevitable regardless of the victim. Deviating from that reference removes the redundant cause and leaves the victim's compliant routing as the remaining but-for link, without the attacker ever acting on the victim's path.

The certificate is computed by the responsibility calculus applied to selector variables that switch each deviation between realised and reference conduct, with the victim's pivotality as the outcome event; counterfactuals are decided by exact world re-execution with citation-chain propagation, so a suppressed trigger silences the conduct that cited it.

What this does not establish. We designed the attacks, the defence and the scoring, and a defence always survives the attacks its author imagined. Three things limit the damage: the attacks are derived from the theorems rather than searched for, so the experiment checks an implementation against a prediction rather than hunting a favourable case; the successful half is reported (both attacks drive a compliant principal's involvement bound to 1.0, which no record machinery prevents); and the certificate is machine-checkable by replaying the two runs it names, so a sceptical reader can verify it without trusting us. Adversaries outside the stated class remain future work, and we claim nothing about them.

S5. Scenario library, planted ground truth, and the analytic oracle

How the families are instantiated. The abstract family definitions below fix the causal structure each scenario must realise; the facility register instantiates them with physical semantics, and this paragraph records that instantiation so the two can be checked against each other. In the facility, principals issue typed commands (`move_to`, `set_speed`, `halt`, `set_param`) carrying declared effect predicates; a deterministic kinematic world advances at 0.1 s ticks; the safety predicate fires at the first tick with human–robot separation below 0.5 m, with severity $\frac{1}{2}mv^2$ at that tick, and the quality predicate at a batch parameter outside its acceptance range. *Redundancy* is two independently sufficient speed-up commands from distinct principals; the *inert twin* is the same record with the second command's actuator locked out, so the two families' transcripts are bitwise identical and differ only in whether an effect propagated; *preemption* is an accused command physically halted before its effect reaches the corridor, with the incident produced by a later command; *omission* is a registered halt duty left unfulfilled; *delegation* is a reckless route relayed by faithful conduits.

The physics oracle and its independence. Ground truth on the facility register is computed by exact world re-execution: for a candidate set the world is re-run from its initial state with those command messages treated as absent, and the declared predicate is evaluated on the resulting trajectory. No language model participates at any point, so this register is immune to the serving-stack nondeterminism that the live-corpus register must manage by protocol. The responsibility profile is then the Chockler–Halpern grade over that outcome function, computed by an analytic implementation written against the definitions and sharing no code with the gate-and-interval engine under test. We exploit this to run a standing cross-implementation check: on the causal layer, where the two share nothing, ground truth and engine must agree. The check is reported in the main text together

with the defect it caught—a gate that retired `halt` commands by kind rather than by production path, producing systematic under-attribution on incidents where a halt caused harm through timing.

Two semantics matter for reproducibility. Interventions act at the consumption level: the record is fixed, and a counterfactual world treats a set of command messages as absent, where a command is world-effective unless it or a policy-chain ancestor is absent. The omission variable inserts the *dutiful* halt obtained by running the certified supervisor policy offline over the recorded sensor reports—a corrected value sourced from the registered duty specification, never imagined. Ground truth is reported in two layers: the causal profile (Chockler–Halpern responsibility over the physics) and the culpable profile (causal responsibility restricted to principals in breach of a content-determined duty). The layers differ routinely and instructively: a compliant route-issuer is often a genuine but-for enabler of a collision, and appears with positive causal responsibility and zero culpability—involvement without breach, which is the groundedness distinction of Theorem S1 made physical. The *sampled* register draws these dimensions independently (redundancy width, preemption, lockout, delegation depth, omission, unit, and the declared predicate) rather than composing a single incident type with parameter noise; only 57.8% of draws are adverse, so structure, not construction, decides whether harm occurs.

Each family below is given by its slice-model composition; all are monotone under the gate convention of Note S1. Ground truth is computed by an *analytic oracle* implementing Proposition S1 directly—prime-implicant extraction followed by minimal-transversal enumeration on the composed outcome function—written against the definitions and sharing no code with the engine under test. Supplementary Figure S1 walks the redundancy family end to end.

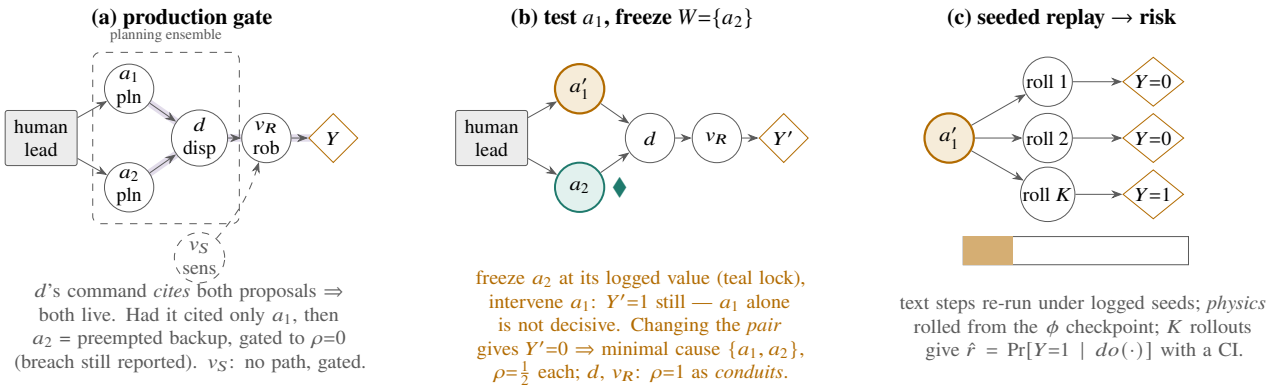
Redundancy (true overdetermination). Planners a_1, a_2 each issue an independently sufficient command; a dispatcher d cites both; the robot r executes. Composed outcome $Y = a_1 \vee a_2$ through conduits d, r. Ground truth: $\rho_{a_1} = \rho_{a_2} = \frac{1}{2}$ (minimal cause $\{a_1, a_2\}$); $\rho_d = \rho_r = 1$ as singleton but-for causes, *conduit-flagged*; no breach findings.

Inert twin. Transcript-identical to redundancy, but the dispatcher's command cites only a_1; a_2's proposal has no production path (and a_2 skipped a required review step). Ground truth: $\rho_{a_1} = 1$; $\rho_{a_2} = 0$ with an exoneration note *and* a breach finding—the family exists to force the breach-without-causation output. Separating redundancy from the inert twin requires the citation structure; transcripts are indistinguishable by construction.

Preemption. a_1 issues a hazardous command; a supervisor halt arrives first and the dispatcher discards a_1's command (no realised effect); a later command a_3 produces Y. Ground truth: $\rho_{a_3} = 1$; a_1 exonerated for Y (gated out, Proposition 2) with a breach finding for the hazardous issuance.

Omission. A sensor records human entry; the duty registry obliges the supervisor s to issue a halt within τ; none is issued (absence variable $\bar{H}_s = 1$); the standing motion command c proceeds and $Y = c \wedge \bar{H}_s$. Ground truth: $\rho_{\bar{H}_s} = 1$ and $\rho_c = 1$ (each conjunct is a singleton but-for cause), breach finding for s only. The family makes explicit that causation and culpability separate: the duty-compliant issuer of c is causally implicated and cleared on breach, which the verdict states rather than hides.

Delegation chain. $u \to p_1 \to p_2 \to d \to$ actuation, with the erroneous transformation introduced at p_1 and relayed faithfully thereafter. Ground truth: every chain member is a singleton but-for cause ($\rho = 1$); p_2, d are conduit-flagged by the faithful-relay test; the breach finding localises to p_1. The family exercises the design position that ρ is an input to judgment: but-for chains grade everyone, and the verdict's flags and breach findings carry the discrimination.



Extended Data Fig. S1 | The causal machinery on an intra-ensemble redundancy incident. Inside the planning ensemble, planners a_1 and a_2 each produce the unsafe plan (either alone suffices); a single dispatcher d emits the one command the robot executes. **(a)** The *production gate* keeps exactly the messages whose signed chain reaches the harm—and here the record is decisive: whether a_2 is a half-responsible joint cause or a preempted backup with $\rho = 0$ is decided by *which proposals d's command cites*, a fact the enforced lineage records and no transcript reconstruction can recover. **(b)** Grading: intervening on a_1 while freezing a_2 at its recorded value leaves the incident intact; only changing the pair flips it, so $\{a_1, a_2\}$ is the minimal cause and each carries $\rho = \frac{1}{2}$—where single-site counterfactual scoring returns zero for both. The dispatcher and robot are but-for causes (ρ=1) flagged as *conduits* by clean-input replay (Supplementary Note 7). **(c)** Counterfactual outcomes are estimated by seeded re-execution plus physics rollouts from the ϕ checkpoints.

Record attack. Any of the above with an attack overlay: forge (inject items failing verification), drop (delete sealed items), reorder, equivocate. Ground truth: the clean-family verdict transformed exactly as Theorem 1 prescribes—bitwise unchanged under forgery, interval-widened under deletion, with $\underline{\rho}$ of fully-certified principals never raised.

Negative controls. **NoCause**: Y is produced by an exogenous environmental event with no principal in the slice; correct output is universal exoneration. **TrivialCause**: a single command with a direct path; correct output is that command's author at $\rho = 1$.

Parameterised generator. Beyond the named families, incidents are sampled by composing monotone gadgets—disjunction of width w (ground truth $1/w$ each), conjunction of width k (each member 1), chains of depth d with a planted deviation, and duty-conditioned absences—into a random series-parallel outcome function, then realising the function as a message topology. The oracle computes ground truth on the composed function; because gadget composition preserves monotonicity, Proposition S1 applies throughout, and the sampled register measures the estimator against an implementation-independent target.

S5.1. Grounding the incident library in public accident data

The register's incident structures are grounded in a corpus of public robot-accident records compiled for this paper (42 entries: OSHA IMIS accident narratives under the keyword "robot," 1984–2024, with 28 detail records fetched individually; NIOSH and state FACE investigation reports; a published analysis of 41 robot-related fatalities in United States CFOI data 1992–2017; and published accident-pattern taxonomies). The mapping from observed causal shapes to register families, with per-shape counts from the corpus, is: single-site command error (11) to preemption and record attack and the sampled register's single-fault draws; omission of a duty-bearing safeguard, lockout not applied, guarding defeated, supervisor absent (11), to omission and the sampled register's

Extended Data Table S1 | Planted ground truth for the named families. Grades from the analytic oracle; flags and breach findings from the family specification. Conduit-flagged grades are reported with the flag attached, per the main text.

Family	Principal	ρ^{gt}	Flags	Breach
Redundancy	a_1, a_2	$\frac{1}{2}, \frac{1}{2}$	—	—
	d, r	1, 1	conduit	—
Inert twin	a_1	1	—	—
	a_2	0	exonerated	review skipped
Preemption	a_3	1	—	—
	a_1	0	exonerated (gated)	hazardous issuance
Omission	s (via $\bar{H}_s$)	1	omission	halt duty
	issuer of c	1	—	—
Delegation chain	u, p_1, p_2, d	1 each	p_2, d: conduit	p_1: deviation
No-cause control	all	0	exonerated	—
Trivial control	author	1	—	(as drawn)

omission axis; unexpected startup during maintenance (7), the modal fatal pattern in the CFOI analysis, to the held-out unexpected-startup family, in which every issued command is duty-compliant and the sole breach is the omitted pre-entry halt; preemption and defeated protections (6) to the inert twin and the lockout axis; mixed conjunctive failures (6) to the sampled register's compositional draws; chain delegation (1) to the delegation chain. Two provenance limits are stated rather than smoothed over. Overdetermination (the redundancy family) is *absent* from the accident corpus: every multi-failure incident in it is conjunctive. Its provenance is the legal and causal-theory literature on overdetermined harm, not accident data, and we keep it because a graded calculus must handle the structure the doctrine treats as hard. Chain delegation appears exactly once in the corpus; its prominence in our register reflects the delegation depth of agent systems, not the frequency observed in industrial records to date.

S6. Canonical form and granularity invariance

A principal should not be able to change its responsibility by re-chunking its own output—splitting one command into three messages or merging three into one. Two mechanisms make this so: the per-principal maximum of Eq. (2) of the main text, and a canonical contraction.

Definition S7 (Same-author conjunctive group)**.** Messages $m_1, \ldots, m_k$ by the same author form a conjunctive group in $\mathcal{M}_S$ if every mechanism depends on them only through the conjunction $m_1 \wedge \cdots \wedge m_k$ (formally: the composed outcome function is invariant under permuting the group and under replacing the group by a single variable equal to their conjunction). The *canonical contraction* replaces the group by one representative variable g.

Proposition S3 (Granularity invariance)**.** *On monotone slice models, canonical contraction leaves every principal's grade unchanged: $\rho_p(\mathcal{M}) = \rho_p(\mathcal{M}/g)$ for all p.*

Proof. By Lemma S3, minimal causes are minimal transversals of the satisfied prime implicants (Proposition S1), so it suffices to show the contraction induces a size-preserving correspondence between minimal transversals touching the group and those containing g.

No minimal cause contains two group members. If $m_1, m_2 \in X^*$, note that setting $m_1 \leftarrow 0$ already sets the conjunction to 0; since mechanisms see the group only through the conjunction, the propagated evaluation of $X^* \setminus \{m_2\}$ equals that of X^*, so X^* was not minimal.

Projection. Let X^* be a minimal cause containing exactly one member m_i. Replacing m_i by g yields a set of equal size that falsifies Y in $\mathcal{M}/g$ (setting $g \leftarrow 0$ has the same effect on every mechanism as setting $m_i \leftarrow 0$), and it is minimal there: a strictly smaller falsifying subset would lift (below) to a strictly smaller falsifying subset in $\mathcal{M}$, contradicting minimality. Causes disjoint from the group are unaffected.

Lift. Let $X^* \ni g$ be a minimal cause in $\mathcal{M}/g$. Replacing g by any single member m_i yields an equal-size set falsifying Y in $\mathcal{M}$ (same effect on all mechanisms), and it is minimal by the symmetric argument.

The correspondence preserves sizes and authorship of the group (all members share the author), so the sets over which each principal's maximum in Eq. (2) ranges have identical grade multisets, and ρ_p is unchanged. □

Remark S10. Even without contraction, Eq. (2)'s maximum absorbs a principal's *own* chunking: if a size-s minimal cause uses one of p's chunks, its grade $1/s$ is what p receives no matter how many sibling chunks exist. The contraction is a normalisation that additionally stabilises the grades of *other* principals whose causes interact with the group, and it is applied before grading in all experiments, and the released test suite checks the invariance on a re-chunked record.

S7. Record substrate and replay details

Storage tiers. The record is held in three tiers: a hot append-only log at the recorder; a warm content-addressed store for payloads (only hashes are sealed, so bulky or personal payloads can be encrypted and crypto-shredded for data protection without breaking the seal); and cold sealed epochs whose Merkle roots are published append-only. Two deployment profiles instantiate the seal: Profile A (single operator) publishes roots to a signed transparency log with periodic external anchoring; Profile B (multi-stakeholder) has a keeper set—operator, vendors, insurer—co-sign each epoch, so no single party can rewrite or fork history. Under either profile the record a reader verifies is the one drawn in Figure 1 of the main text: hash-chained blocks, each sealing an epoch of signed messages under a Merkle root.

Equivocation. Presenting different histories to different parties is defeated by the single sealed sequence: receipts and epoch commitments bind every consumer to the same root, and divergent roots are themselves cryptographic evidence of recorder misbehaviour, surfacing in the suspect set.

Replay stacks and action-match. The pinned stack fixes model weights, decoding parameters, seeds, tool versions, and the simulator build, making factual replay bit-exact and interventional replay deterministic. The hosted stack replays against components that cannot be pinned; certification then reports the confidence interval of Proposition 1 together with an *action-match* score: under an order-preserving alignment of the replayed and recorded actuation sequences, the fraction of aligned events agreeing in type, target, and parameters within declared tolerances. A low action-match flags that the replayed system is no longer the recorded system, independent of the risk estimate.

S8. Reproducibility

The reference implementation is a Python package organised as `record/` (accountable messages, verification, Merkle sealing), `engine/` (gate, grade, certify), `verdict/` (analytic responsibility, intervals, canonical form), `facility/` (incident families, generator, analytic oracle), `corpus/` (live pipeline, replay oracle), `baselines/`, and `metrics/`, with one registered runner per experiment identifier under `scripts/`. Each runner writes a JSON artifact under `results/` carrying the configuration, per-incident outputs, and summary statistics, and every

run sweeps generation seeds $k \in \{0, 1, 2\}$ where the register is stochastic. All numbers, figures, and tables in the main text are regenerated from those artifacts by scripts included in the release, and a verification script re-checks every reported quantity against its artifact before submission. The simulation layer is CPU-deterministic; pinned-stack replay of open-weight planners uses seeded deterministic serving. Experiments ran on NVIDIA A100-SXM4-80GB GPUs under Python 3.11, NumPy 2.4, and SciPy 1.17, with open-weight models served by vLLM 0.14.1; the archived release pins exact package versions and the container digest.

S9. Registered disclosures and measurement history

This note consolidates, in one place, every deviation from registration, every refuted prediction, and every measurement whose history bears on how much weight it deserves. The experiment log in the released repository records each item with its registration date.

Refuted predictions (six). (i) The regime decomposition predicted that 28–42% of live-corpus incidents would exhibit joint solver–aggregator causation; the measured share is 95–98% at 72B. (ii) The volume-confound prediction expected judge attributions to track message volume; under a fair manipulation the measured slope is $\beta = 0.000$. (iii) The completeness-barrier bite prediction expected ground-truth access to confer a measurable advantage on honest errors; the paired difference spans zero. (iv) The repair-curve prediction expected unranked arms to stay flat across budgets; they brute-force upward. (v) The Shapley baseline was predicted to recover the redundancy split and to land in 0.30–0.55 on the live register; it does neither, erring 0.833 on redundancy while attaining 0.223 live, which makes it the strongest baseline in the paper rather than a weak one. (vi) Shapley was predicted to stay above AUDITA in every model family; on Qwen2.5-7B it is below (0.041 versus 0.111, paired 95% CI $[-0.141, -0.004]$), the one register in this paper where a baseline is more accurate than the graded verdict. Each refutation is reported in Results with the analysis it forced.

Measurement history of the cross-family argmax cell. The Llama-3.1-8B cell first yielded 13 scoreable incidents, on which the judge led argmax culprit identification 1.000 to 0.615. Because extending collection after an unfavourable result is optional stopping, we fixed in advance of the extension that the enlarged cell would be reported in whichever direction it moved, with no change to arms, metrics, or exclusion rules. The extension to 25 incidents (offsets disjoint from the original draw, labels serially re-verified, 139/142 reproduce) moved the gap from thirty-eight points to twelve (0.960 versus 0.840). Both estimates appear in the log; the main text reports the extended cell together with this history's existence. Readers preferring the conservative reading should treat the small first sample as the less reliable of the two, which is also what its width implies.

Amended designs (two). The second-family experiment was registered for Llama-3.3-70B and amended to a matched-scale contrast (Qwen2.5-7B versus Llama-3.1-8B) when cluster capacity made a second 70B unservable on four dated attempts; the amendment isolates family at fixed scale and adds a within-family scale arm. The external-benchmark experiment was registered for Who&When Pro and executed on Who&When when Pro's data and code remained unreleased at run time; no Pro number appears anywhere in this paper.

Instrument defects caught before any number reached the paper (seven). A kind-filter in the production gate that excluded halt commands (caught by the engine-versus-oracle independence check); a tautological residual design; a repair operator that set rather than clamped speeds; a tie-break that under-scored a baseline; a budget-unfair random control; a rigging construction whose victim was not in fact compliant; a duty-sensitivity population that could not span the registered standards; and a physical-path-only production gate that missed

citation-created production (surfaced by the level-2 construction; after repair, full re-runs of the principle-family and record-attack experiments reproduced every previously published mean exactly); a judge cell whose candidate list omitted the silent duty-holder, making the correct answer unnameable (the cell was rerun in two registered variants, actor-only and duty-informed, and both are reported); and a first-version geometry for the unexpected-startup family in which the worker walked into the halted robot, so the omission was not a cause and a compliance check passed vacuously (the family was re-parameterised so the derived halt averts the harm, the keep condition was made non-vacuous, and every cell was rerun). Each was repaired and re-registered before its experiment's numbers were accepted.

S10. Extended related work

The main-body Related Work covers the lines closest to AUDITA: failure attribution in multi-agent systems, actual causality and graded responsibility, and the manipulation-of-attribution literature that flanks our blame-shift certificate. This note provides the extended survey across the remaining lines the project engaged, organized by the role each plays relative to AUDITA.

Strategic and adversarial manipulation of attribution

A small adjacent literature studies attribution when the attributed parties are strategic, and none of it, to our knowledge, addresses the framing problem our guarantees target. In cooperative sequential decision making, game-theoretic blame attributions provably misalign incentives (non-performance-incentivizing Shapley, over-blaming Banzhaf), motivating attribution methods with better structural properties under *uncertainty*, but the agents there do not attack the attribution of *others*[42]. Recent work on retrospective counterfactual responsibility in concurrent stochastic games lets agents trade their *own* expected responsibility against reward in equilibrium, forward-looking strategy synthesis, with no adversarial manufacture of a victim's responsibility and no evidence layer[43]. In the explanation literature, adversarially crafted models can fool perturbation-based feature attributions such as LIME and SHAP[44]; the object manipulated there is a model explanation, not a principal's graded responsibility over a certified record. Accountability systems in the PeerReview tradition guarantee that correct nodes are never exposed, but detect *protocol deviation* against a deterministic reference implementation[45]; our Attack I is protocol-compliant conduct, invisible to deviation detection by construction. Our blame-shift accountability theorem differs from all four lines in both question and machinery: it asks whether a coalition can raise a *victim's* responsibility, answers by applying the Halpern–Pearl test at the meta level (deviations as causes of the victim's pivotality), and returns a replay-checkable certificate that grades the framers. On the barrier side, concurrent work establishes statistical limits of ground-truth-free auditing, worst-case calibration error of any label-free estimator in the rare-error regime[46]; our completeness barrier is the per-instance, adversarially-relevant counterpart: a logical soundness/completeness contradiction for content-determined breach standards, tied to negligence semantics rather than to estimator calibration.

Causation in law, defaults, and omissions

The legal doctrine AUDITA answers to is the NESS test, a cause as a necessary element of a sufficient set of actual conditions[17] , and the broader treatment of causation in the law by Hart and Honoré[18], from which the verdict's duty–breach–causation–harm structure is drawn. The philosophical literature distinguishes dependence from production as two concepts of causation[47], precisely the distinction our gate-and-grade test operationalizes; graded causation with defaults and normality gives omissions their standing as causes[48], which our duty-indexed absence variables implement mechanically. Empirically, human causal judgments track counterfactual simulation[49], and

human responsibility intuitions are sensitive to factors, such as how critical or salient a contributor appears, that a defensible audit must not inherit; our volume-confound experiment tests the machine analogue of exactly this failure.

The responsibility gap and AI accountability

That learning systems open a gap between harm and accountable agent is the seminal observation of Matthias[50], refined into four distinct gaps, culpability, moral accountability, public accountability, and active responsibility[51], and rooted in the older problem of many hands[52]. AUDITA targets the culpability and public-accountability gaps by manufacturing the evidentiary object they presuppose. Governance work on agent visibility proposes identifiers, real-time monitoring, and activity logs[53]; authenticated-delegation frameworks issue scoped, auditable authority from humans to agents[54]; and on-chain identity registries are commoditizing agent identity at scale[55], AUDITA consumes these primitives (its delegation certificates and duty clauses are exactly such tokens) and adds what they do not attempt: post-hoc causal attribution over the recorded conduct. System- and process-level AI auditing assesses documentation, monitoring, and governance of deployed systems[56,57]; regulation demands event recording and traceability[8] within safety regimes that presume incidents can be reconstructed[9,10]. AUDITA audits the event rather than the system, and proposes what the mandated records must *be* for those obligations to purchase accountability.

Accountability systems and tamper-evident records

The distributed-systems ancestry of the record is direct. PeerReview established that nodes signing all messages into tamper-evident logs, audited by witnesses, yields two guarantees, detected faults are irrefutably linked to a faulty node, and correct nodes can always defend against false accusation[45]; Theorem 1 is our generalization of the second guarantee from protocol deviation to graded semantic responsibility. Accountable Virtual Machines extended the recipe to record-replay-blame over full executions[58], and its stated limits are exactly our deltas: it requires a deterministic reference implementation and cannot fault faithful execution of flawed software, language-model principals have no reference implementation, are nondeterministic, and act on the physical world. CSAR made accountability survive randomness by logging application inputs and random choices[59], the ancestor of our seed-and-model-hash fields; formal definitions of accountability via judge-rendered verdicts, and their relation to verifiability, come from Küsters, Truderung and Vogt[60]. The data structures beneath the seal are the tamper-evident history trees of Crosby and Wallach[61], fork consistency against equivocating servers from SUNDR[62], forward-secure audit logs[63], Merkle commitments[23], and transparency logging at Internet scale[24], under classical Byzantine and network-adversary models[29,30]. Practitioner standardization is beginning: an IETF draft specifies hash-chained, optionally signed audit records for agents with an explicit mapping to the AI Act's logging article[64], a single-writer schema without lineage, receipts, or physical binding, whose spirit our record extends. The nearest new relative is the verifiable-transcript line, which applies SUNDR-style signed digests and fork detection to LLM conversation transcripts under an untrusted-provider threat model[65]; it certifies a two-party transcript, whereas AUDITA certifies a multi-principal command graph with authored lineage, actuation binding, and attribution on top.

Agent observability tooling

Industry telemetry for agents is consolidating around vendor-neutral semantic conventions for agent, tool, and model spans[66], on the pedigree of large-scale distributed tracing[67]; academic work taxonomizes what agent operations should trace[68] and how agentic systems can be observed and optimized in operation[69]. The gap this cluster leaves is the one AUDITA fills: telemetry captures what happened without identity, integrity, or assessment,

no field binds a span to a signing principal, nothing proves the trace complete, and no semantics turns spans into responsibility. Our message format is deliberately close to these conventions so that the three fields standard logs lack (authored citations, committed effects, pinned seeds) read as an extension rather than a replacement.

Threats to agent and robot collectives

The attack literature supplies both motivation and the record-corruption manipulations used in our evaluation. Jailbreaking attacks produce harmful physical actions on deployed LLM-controlled robots[70]; in LLM-coordinated multi-robot teams, compromising a single entry robot propagates malicious intent through peer communication, with obedience reaching 1.00 and full-team compromise in as few as three rounds in the strongest reported cases[71], the staged version of the incidents AUDITA investigates. Surveys of agent communication protocols document tool poisoning, injection, and cross-boundary provenance gaps[72]. Prevention-oriented security frameworks harden the channel: the Aegis Protocol combines decentralized identifiers, post-quantum signatures, and zero-knowledge policy compliance under an extended Dolev–Yao adversary[73], and BlockA2A anchors agent-to-agent interoperability in verifiable identity[74]. These are complements, not competitors: they aim to prevent the incident, AUDITA to adjudicate it, and a hardened channel makes the certified record's assumptions easier to discharge.

Safety science and incident investigation

The incumbent methodology for accident analysis is systems-theoretic: STAMP models accidents as control-structure failures and CAST analyses them deliberately blame-free, asking why and how rather than who[75]. AUDITA is designed to feed, not replace, that practice: the same certified record supports a blame-free learning analysis and the evidentiary who-and-how-much account that liability, insurance, and regulation additionally demand. The ethical black box argued that robots need flight recorders specified for accident investigation[76]; our record is its multi-principal, agentic-era successor, and it attributes as well as logs. Adjacent blockchain-robotics work uses ledgers for swarm coordination and Byzantine defence, token economies that neutralize harmful robots[77], reviewed for mobile multi-robot systems in the Nature portfolio[78]; that line secures coordination at run time, while AUDITA uses ledger machinery for post-incident causal attribution.

Neighbouring methodological fields

Three method families adjoin ours and mark its boundary. Root-cause analysis for microservices computes type-level causal effects over inferred dependency graphs, increasingly with partial identification under latent confounding and robustness to graph misspecification; our problem is token-level actual causation over a graph that is signed by construction rather than inferred, with uncertainty that is adversarial rather than statistical, and with graded responsibility, exoneration, and an innocence guarantee that root-cause outputs do not carry. Provenance-based forensics captures whole-system operating-system provenance and reconstructs attack stories over it[79], the mature syscall-level analogue of our problem, without message semantics, duties, or embodiment. Causal statistical fault localization pioneered causal inference over program dependence graphs[80], and its slicing-to-the-cause-effect-chain pattern prefigures our production gate, ported here to signed cross-principal command graphs with physical outcomes. Finally, counterfactual credit assignment in cooperative multi-agent reinforcement learning[81,82] shares the counterfactual core but serves training-time reward shaping; post-hoc evidentiary attribution differs in object (a concrete incident), in adversary (the record itself is attacked), and in consumer (parties with legal stakes), which is why the game-theoretic attributions it favours misbehave as responsibility measures[42].

The deployment substrate

The systems whose incidents AUDITA is built to audit are documented in the main text: language-grounded robot planners and vision-language-action models [1,2,83], open-ended and communicative agents [84,85], and orchestration frameworks [6,7], evaluated on field-standard task suites [86–88] whose shapes our facility simulation abstracts. Across all these clusters, the pattern is the one the paper argues: each ingredient of AUDITA has deep roots in its home field, and what is new is the composition, certified evidence, production-gated graded causation, and replay verification bound into one pipeline whose output is designed to survive a hostile audience.