

Who Can Undo the Agent?

Redressability as a First-Class Safety Property for Tool-Using AI Agents

Saad Alqithami and Abdulkareem Alzahrani

Department of Computer Science, Al-Baha University
Al-Baha 65779, Saudi Arabia
{salqithami;ao.alzahrani}@bu.edu.sa

Abstract

Tool-using AI agents increasingly act inside institutional workflows by updating records, sending notices, changing account status, scheduling services, and triggering review processes. Existing agent-safety evaluations largely ask whether agents avoid harmful actions. Deployed institutions must also determine whether affected people and responsible reviewers can discover, understand, contest, reverse, and obtain remedy for consequential actions after they occur. We operationalize redressability, a property grounded in legal doctrine and accountability scholarship, as a first-class safety property of tool-using agents. The framework defines eight scoreable dimensions: detectability, traceability, contestability, reversibility, remedy, recurrence prevention, burden allocation, and value-chain assignability. We instantiate the framework in AGENT REDRESSBENCH, a benchmark of 40 seed and 24 stress scenarios; REDRESS TRACES, an action-level provenance schema; and REDRESS GATE, a proposed runtime-enforcement pattern. Diagnostic experiments with GPT-4.1 and Llama 3.1 8B show that guardrails, trace generation, and escalation address distinct failure modes. Guardrails reduced automatic unsafe-action labels without creating redress evidence, whereas trace contracts reliably elicited accountability artifacts. Trace-only outputs nevertheless received unsafe-action labels in 26/40 Llama seed cases, sometimes while documenting that required approval had not been obtained. A blinded single-validator assessment of all 96 GPT-4.1 stress trajectories rated 93 trajectories safe and three ambiguous, while sharply separating trace from non-trace conditions in redressability: mean scores were 19.21/24 and 19.33/24 for trace and trace plus escalation, compared with 4.38/24 and 4.58/24 for baseline and guardrailed conditions. These findings support the separation of substantive action safety from procedural redressability and motivate runtime enforcement as a design implication rather than an evaluated treatment.

Code, Data, and Supplementary Materials —
<https://github.com/alqithami/redressbench>

Introduction

Large language model systems are shifting from chat interfaces toward agents that act through browsers, code execution environments, messaging systems, workflow tools, databases,

Copyright © 2026, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

and APIs. Recent agent-safety benchmarks emphasize this shift by evaluating agents in multi-turn tasks with real or realistic tools rather than isolated text outputs (Tur et al. 2025; Vijayvargiya et al. 2026; Shapira et al. 2026). The object of safety analysis is therefore no longer only a model response but a socio-technical trajectory: a model receives instructions, consults records, invokes tools, changes institutional state, and creates consequences for people who may not have interacted with the model at all.

Most agent-safety evaluations ask a prevention-centered question: did the agent take an unsafe action, refuse a harmful request, or violate a policy? Prevention is necessary, but it is incomplete. In deployed institutional settings, the relevant question often arrives after an action has already occurred: can the affected person find out what happened, see the basis for the action, challenge it, reverse it, and obtain a meaningful remedy? An agent that avoids obvious policy violations may still be unsafe if it silently closes a public-benefits case, updates an employment record, denies a refund, or modifies a waitlist without creating evidence that a person or institution can use to contest the outcome.

This paper develops REDRESSABILITY as an evaluation target for tool-using agents. The term itself is not new. Redressability is an established element of standing doctrine in United States federal courts, where a plaintiff must show that a favorable decision would likely redress the claimed injury (*Lujan v. Defenders of Wildlife* 1992), and the underlying idea that people subject to institutional power must retain a usable path to repair runs through technological due process, contestability, reviewability, and algorithmic recourse (Citron 2008; Lyons et al. 2021; Cobbe et al. 2021; Kaminski and Urban 2021; Wachter et al. 2018). Our claim is therefore not conceptual invention. It is that this familiar property becomes newly measurable, and is currently unmeasured, when the acting entity is an AI agent executing tool calls. We define redressability as the degree to which affected people and responsible institutions can discover, understand, contest, reverse, and remedy consequential actions taken by an AI agent. The construct is related to, but distinct from, transparency, explainability, recourse, robustness, and refusal behavior. A redressable agent need not expose hidden chain-of-thought or private system prompts. It must preserve enough action-level evidence to make institutional review possible.

Motivating example. Consider a workplace agent asked to

restrict an employee’s building access after a manager reports misuse. A prevention-centered evaluation asks whether the agent disables access when the evidence is insufficient. A redressability evaluation asks additional questions even when access restriction is substantively justified: Did the agent record the evidence handles? Did it identify the policy basis? Did it send or schedule notice? Is there an appeal route? Can access be restored if the record is wrong? Who is accountable: the model provider, the agent builder, the HR system, the deploying employer, or the human requester? These questions are not secondary details. They determine whether the action can be institutionally corrected.

Contributions. The paper makes four contributions.

1. We consolidate redressability, adapting a concept with deep roots in legal and accountability scholarship, into an operational, first-class agent-safety property, and separate it from harm prevention, explanation, and ordinary logging. The contribution is the transfer and operationalization of the concept for tool-using agents, not its invention.
2. We introduce AGENT REDRESSBENCH, a benchmark with 40 seed scenarios and a 24-scenario stress split across workplace, public-service, healthcare-admin, and customer-support domains.
3. We propose REDRESS TRACES, a minimal action-level provenance schema, and derive REDRESS GATE, a runtime-gated design pattern motivated by observed procedural-trace failures. The gate is a design implication and is not evaluated as an experimental treatment in this study.
4. We report diagnostic experiments with GPT-4.1 and Ollama/Llama 3.1 8B and a blinded single-validator assessment showing that guardrails, trace generation, and escalation address empirically distinct failure modes.

The goal is not to claim that the current benchmark proves real-world deployability. The goal is to make post-action accountability measurable enough that agent evaluations can test it directly.

What this paper does not claim

The paper deliberately narrows its claims in four ways. First, redress traces do not make an agent safe. The Llama 3.1 8B results show that a model can generate plausible trace fields while still taking an unsafe action. Second, not every consequential action should be refused. Many institutional workflows require action, accompanied by notice, evidence, appeal, rollback, and remedy when rights or interests are affected. Third, a structured trace does not create accountability by itself. Accountability also requires organizational commitments, review capacity, and authority to reverse or compensate. Fourth, the diagnostic automatic scorer is not a final judge. It is a high-recall instrument for identifying cases that need human review.

These non-claims matter for positioning the work. A redressability benchmark should not reward performative paperwork, and it should not collapse governance into refusal. The scientific target is a more precise question: when an agent acts, does the surrounding system preserve the procedural

conditions under which that action can be challenged and repaired?

Problem Setting

Terminology: AI systems, agents, and tools

Definitions of “agent” vary and are actively debated (Chan et al. 2023; Shavit et al. 2023), so we fix terminology before defining the property. We use *AI system* in the broad regulatory sense of the EU AI Act: a machine-based system that infers from its inputs how to generate outputs such as predictions, content, recommendations, or decisions that can influence physical or virtual environments (EU AI Act 2024). A *tool* is a callable interface that a deployer exposes to such a system: an API endpoint, database operation, messaging function, or workflow action whose invocation reads or, critically, writes state outside the model’s own context. A *tool-using AI agent*, the object of this paper, is an AI system, here built around a large language model, that (i) receives a natural-language objective, (ii) selects and sequences tool calls in pursuit of that objective, and (iii) is permitted to execute side-effective tool calls under delegated authority without per-action human review. “Agentic AI” is often used as an umbrella label for the same class of systems. Following prior work, we treat agency as a matter of degree along dimensions such as goal complexity, environmental impact, and independence of execution (Chan et al. 2023; Shavit et al. 2023). Nothing in our framework depends on where a system falls on that spectrum: redressability attaches to any AI system that executes consequential tool calls. A system that only drafts recommendations for a human to execute raises related but distinct questions, because the executing human remains the proximate accountable actor, so the benchmark scopes to systems that execute the call themselves.

Setting

We study agents that can perform *consequential tool calls*: tool invocations that can change institutional state or materially affect an identifiable person. Examples include freezing an account, closing a case, changing a waitlist status, creating an HR record, denying a refund, or sending a notice. Each trajectory contains a task context, policy text, records, permitted tools, a user request, an affected party, and the agent’s output.

Let a trajectory be $\tau = (x, a_{1:T}, y)$, where x is the task context, $a_{1:T}$ is the sequence of proposed or executed tool actions, and y is the structured output. Let $C(a_t)$ indicate that action a_t is consequential. A safety evaluation usually estimates whether some a_t is harmful or prohibited. A redressability evaluation estimates whether, for each consequential action, the trajectory creates a usable path for institutional review and repair.

We distinguish four action outcomes:

- **Wrongful harm:** the agent takes an adverse action that policy or evidence does not support.
- **Valid consequential action:** the agent takes a high-impact action that is substantively supported.
- **Safe escalation:** the agent preserves evidence and routes final authority to a human or institutional reviewer.

- **Unredressable consequential action:** the action may be valid or invalid, but the trajectory lacks adequate notice, traceability, contestability, reversibility, or remedy evidence.

The fourth category is the paper’s focus. Many institutional harms do not arise only because the substantive action was wrong. They arise because no one can later reconstruct, contest, or correct the action.

Threat model

We assume the agent is deployed by an institution and receives delegated authority to use a bounded set of tools. The user request may be benign, mistaken, incomplete, self-interested, or adversarial. Records may be stale, partial, or disputed. Policies may require human approval, affected-party notice, or an appeal route before adverse action. The affected person may not be the immediate user, and the tool runtime faithfully executes any syntactically valid tool call unless an external gate blocks it. The main risks are therefore not only malicious misuse but also overconfident automation, premature escalation of administrative sanctions, invisible record changes, and post-hoc inability to assign responsibility.

This threat model leads to three evaluation requirements. The benchmark must include tool calls with side effects, distinguish valid consequential action from wrongful harm, and score whether the trajectory preserves redress evidence even when the final outcome is substantively valid.

Related Work and Gap

Agent safety and tool use. Recent benchmarks evaluate safety risks that emerge when LLMs use tools in realistic settings. SafeArena focuses on harmful web-agent tasks across websites and harm categories (Tur et al. 2025). OpenAgentSafety evaluates multi-turn, multi-user agent tasks using tools such as browsers, file systems, code execution, terminals, and messaging platforms (Vijayvargiya et al. 2026). ToolEmu emulates risky tool execution in a sandbox (Ruan et al. 2024). AgentHarm measures harmful task compliance (Andriushchenko et al. 2025). τ -bench evaluates tool-agent-user interaction under domain rules (Yao et al. 2025), and security analyses of web-use agents emphasize prompt injection, unsafe web actions, and tool-mediated risk (Shapira et al. 2026). On the governance side, proposals for visibility into AI agents call for agent identifiers, real-time monitoring, and activity logging (Chan et al. 2024), and incident-analysis frameworks extend post-hoc investigation to agent-caused harms (Ezell et al. 2025). This work is essential, but it usually treats the main evaluation target as unsafe action, compliance, or population-level oversight. We ask what evidence remains, for a particular affected person, when the agent acts (Table 1).

Accountability, auditing, documentation, and incidents. Accountability theory describes a relation in which an actor must explain and justify conduct to a forum that can ask questions, pass judgment, and impose consequences (Bovens 2007). Systematic reviews adapt this actor–forum structure to algorithmic systems (Wieringa 2020), and recent conceptual work analyzes how goals, obligations, and sanction mechanisms combine in AI accountability regimes (Novelli et al.

2024; Cooper et al. 2022). Algorithmic auditing frameworks operationalize parts of this relation through end-to-end processes, documentation, and internal review (Raji et al. 2020; Mökander et al. 2024), and empirical work shows that record-keeping choices themselves reshape oversight (Chappidi et al. 2025). Incident-reporting work motivates shared memory of deployed AI failures (McGregor 2021). Risk-management frameworks such as NIST AI RMF emphasize governance, mapping, measurement, and management of AI risks (Tabassi 2023). The EU AI Act makes several of these expectations legally binding for high-risk AI systems: providers must operate risk-management systems (Art. 9), ensure automatic event logging (Art. 12), monitor post-market performance (Art. 72), and report serious incidents (Art. 73), and affected persons acquire a right to an explanation of the role of the AI system in certain individual decisions (Art. 86) (EU AI Act 2024). Unlike the voluntary NIST framework, these obligations are enforceable and backed by administrative penalties (Art. 99). AIES work on model-risk catalogs and harm mapping similarly argues that evaluation should connect system attributes to hazards and harms (Rao et al. 2025; Rismani et al. 2025). Our contribution relative to this line is to make post-action redress a measurable, trajectory-level property of agents: redress traces are the trajectory-level substrate a deployment would need to discharge logging, incident, and explanation duties of this kind.

Due process, contestability, and recourse. The normative lineage of redressability is older than AI safety. Technological due process argued that automated administration erodes notice and hearing rights unless procedural protections are rebuilt into the technology (Citron 2008; Citron and Pasquale 2014). Reviewability frameworks specify the records that make automated decision-making amenable to meaningful review (Cobbe et al. 2021), and legal scholarship has articulated a right to contest AI decisions (Kaminski and Urban 2021). Work on contestability argues that affected people need more than explanations: they need meaningful ways to challenge decisions (Lyons et al. 2021). Counterfactual-explanation and algorithmic-recourse research shifts explanation toward actionability for the affected person (Wachter et al. 2018; Karimi et al. 2022). Procedural-justice research supplies the empirical foundation: people’s judgments of institutional legitimacy depend strongly on process, voice, and acknowledgment, not only on outcomes (Lind and Tyler 1988; Tyler 2006). Sociotechnical critiques warn that purely technical abstractions can miss institutional context and power (Selbst et al. 2019). We extend this line by operationalizing its demands for tool-using agents: a trace must identify not only the model’s stated rationale, but the tool action, evidence handles, policy basis, affected party, appeal route, rollback path, remedy options, and responsible actor stack.

Agents, law, and distributed responsibility. AI-agent legal scholarship emphasizes that technical agency and socio-legal agency do not align automatically. Direct execution of tasks raises issues of delegation, responsibility, authority, and liability (Riedl and Desai 2025). In United States standing doctrine, redressability already names the requirement that a court be able to remedy the injury before it will hear the claim (*Lujan v. Defenders of Wildlife* 1992). We transpose

the same question from courts to agent deployments. Because agents are built from models, tools, data sources, and deployment scaffolding supplied by different actors, responsibility is distributed across an AI supply chain (Hopkins et al. 2025), and philosophers have catalogued the resulting responsibility gaps (Santoni de Sio and Mecacci 2021). The value-chain assignability dimension below is our operational response. Redressability provides an evaluation lens for these issues without reducing them to legal compliance alone.

Redressability Framework

We define redressability along eight dimensions (Table 2). Each dimension is scored on a 0–3 ordinal scale: 0 absent, 1 nominal or generic, 2 procedurally usable but incomplete, and 3 institutionally actionable. The unweighted diagnostic score is

$$R(\tau) = \sum_{d \in D} r_d(\tau), \quad r_d(\tau) \in \{0, 1, 2, 3\}, \quad (1)$$

where D is the set of eight dimensions. The maximum diagnostic score is 24. The unweighted sum is a diagnostic convenience only: we report and interpret dimension-level scores throughout, and the dimensions divide by locus of control between the agent trajectory and the surrounding institution (Table 5). The blinded validation uses the same dimensions and supersedes the automatic scorer when procedural meaning requires judgment.

Three design choices are important. First, redressability is not the same as chain-of-thought transparency. The trace preserves action-level provenance and institutional review information, not private model reasoning. Accountable process does not require resolving model inscrutability (Kroll 2018). Second, redressability is not the same as refusal. A redressable agent may take a valid consequential action, but it must do so through an accountable workflow.

Third, the dimensions deliberately track a conception of redress that is wider than reversal plus compensation. In human-rights and administrative practice, redress spans restitution, compensation, rehabilitation, satisfaction, which includes acknowledgment of fault and apology, and guarantees of non-repetition (UN General Assembly 2005), and access-to-remedy frameworks require grievance mechanisms that are legitimate, accessible, predictable, and equitable (United Nations 2011). Accountability theory adds that a forum must be able to impose consequences, including sanctions, on responsible actors (Bovens 2007). The dimensions encode this breadth. The remedy dimension scores satisfaction-type remedies such as apology and corrected records alongside material compensation. Recurrence prevention corresponds to guarantees of non-repetition. Value-chain assignability makes sanctioning and corrective duties addressable to a specific actor, such as a model provider, agent builder, tool provider, deployer, or human approver, rather than to an undifferentiated system. Procedural-justice research gives an empirical reason to take the non-material components seriously: perceived legitimacy depends on process, voice, and acknowledgment, not only on outcomes (Lind and Tyler 1988; Tyler 2006).

Benchmark and Conditions

AGENT REDRESSBENCH contains synthetic institutional scenarios rather than scraped real cases. This choice avoids privacy risks and allows controlled variation in evidence quality, policy language, affected-party rights, and permitted tools. Version 0.3 contains 40 seed scenarios: 10 workplace, 10 public-service, 10 healthcare-admin, and 10 customer-support/platform-governance scenarios. Version 0.4 adds 24 stress scenarios: six per domain. Stress scenarios increase institutional pressure and include more cases where consequential action is substantively valid but still requires notice, appeal, rollback, and remedy.

Each scenario includes the fields in Table 3: domain, agent role, permitted tools, policy snippets, relevant records, user request, affected party, privacy constraints, unsafe target actions, expected safe action, and hidden risk label. The agent receives the task context but not the hidden labels. The benchmark evaluates four conditions:

1. **Baseline:** complete the request using permitted tools and records.
2. **Guardrailed:** avoid adverse high-impact action under ambiguity, stale evidence, disputed records, accommodation-sensitive information, or missing approval.
3. **Trace:** complete supported actions and emit a structured redress trace for consequential action.
4. **Trace + escalation:** emit redress evidence and escalate ambiguous high-impact cases to a human or institutional reviewer.

Version 0.3 introduced condition-specific output contracts: baseline and guardrailed outputs are not asked to fill the redress schema, while trace conditions must. This keeps the baseline uncontaminated by redress-field instructions and makes the separation between safety guardrails and redress instrumentation visible.

Scenario design principles

The benchmark is built around four design principles. *First, consequentiality:* each scenario includes at least one plausible tool action that can affect an identifiable person or account. *Second, evidentiary variation:* records can be confirmed, ambiguous, stale, contradicted, accommodation-sensitive, or policy-constrained. *Third, procedural specificity:* policies specify not only whether an action is allowed, but what process must surround it. *Fourth, separability:* the hidden labels distinguish wrongful action, valid consequential action, safe escalation, and unredressable action, so the benchmark does not reduce all high-impact action to harm.

Deployment assumptions and institutional affordances

Redressability is partly institutional, so scores are interpretable only relative to an explicit assumption about what the surrounding organization provides. AGENT REDRESSBENCH holds this environment fixed and states it in the scenario materials. Every scenario supplies four things in the agent’s context. First, it supplies the applicable policy

Research line	Typical evaluation question	What it captures well	Gap addressed here
Agent-safety benchmarks	Did the agent perform a harmful or prohibited action?	Tool-mediated risk, harmful compliance, refusal failures	Whether affected people can contest and repair actions after execution
Auditing and risk management	Did the organization document and govern the system?	Lifecycle governance, accountability processes	Trajectory-level evidence for a particular consequential action
Explainability and recourse	Can the person understand or change the decision basis?	User-facing explanation and actionability	Tool provenance, rollback, remedy, and value-chain responsibility
Incident reporting	What went wrong after deployment?	Shared failure memory and post-market learning	Pre-specified redress evidence before an incident becomes public
Accountability theory	Must the actor answer to a forum that can judge and impose consequences?	Actor–forum relations, justification, sanctions	Trajectory-level evidence that lets a forum exercise judgment over a specific agent action
This paper	Can the action be discovered, attributed, contested, reversed, and remedied?	Post-action accountability for tool-using agents	Connects agent trajectories to institutional redress mechanisms

Table 1: Positioning redressability relative to adjacent evaluation traditions.

snippets, including any evidence thresholds, approval requirements, notice duties, appeal routes, and named review owners. Second, it supplies a permitted-tool list that includes the procedural instruments those policies presuppose: notification, record annotation, escalation or review-routing, and, where the scenario contemplates reversibility, rollback-capable operations. Third, it supplies records sufficient to identify the affected party. Fourth, it supplies privacy constraints. In other words, the benchmark assumes a procedurally mature institution: appeal procedures exist, review bodies are named, notification channels work, delegated authority is defined, and rollback mechanisms are available where policy expects them. The v0.3/v0.4 scenarios do not model institutions that lack these affordances, and the agent is never asked to infer an affordance that the scenario withholds.

This assumption fixes the interpretation of the institutionally embedded dimensions (contestability, reversibility, and remedy) relative to the more directly technical ones. A low score on these dimensions in our experiments means the trajectory failed to invoke, attach, or document affordances that the environment made available (for example, executing an adverse action without citing the appeal route that the policy snippet names), not that the deployment lacked the machinery. Conversely, the benchmark does not measure whether a real institution would honor an appeal, staff its review queue, issue an apology, or pay a remedy. Those are properties of the deployment, outside any trajectory’s control. Table 5 makes the division explicit. Practitioners adapting the benchmark to a deployment whose affordances are genuinely absent should score the corresponding dimension against the deployment, as a governance gap, rather than against the agent, and should report the two failure types separately rather than letting them blur inside the aggregate score.

Redress Traces and Runtime-Gated Design

REDRESS TRACES is a minimal action-level provenance schema. It records the action ID, tool call, affected party, evidence handles, policy basis, approval status, reversibility status, contestability route, remedy options, recurrence-prevention step, burden allocation, value-chain actor stack, and privacy-minimization note. The schema uses evidence handles rather than raw sensitive records.

The current experiments test traces as prompt-level output requirements. The Llama 3.1 8B results show why this is insufficient. A model can produce a plausible redress trace while still executing an adverse action whose own trace says that required approval was not obtained. This motivates REDRESS GATE, a runtime mechanism between the model and tools. The gate checks whether a proposed side-effectful tool call satisfies redress predicates before execution (Table 6).

Figure 1 sketches the proposed enforcement rule: nonconsequential calls proceed under ordinary tool validation, while consequential calls require the runtime to evaluate redress predicates over the trace and the institutional policy before execution. Missing or inconsistent predicates convert the action into a blocked-call record and an escalation request.

This design yields a falsifiable hypothesis for subsequent evaluation: runtime-gated redress should reduce procedural-trace failures relative to prompt-only trace conditions while preserving more valid consequential action than blanket refusal or indiscriminate escalation. The present experiments evaluate the evidence motivating the design, not the gate itself.

Experimental Protocol

The empirical study has two components. First, diagnostic model runs test the benchmark pipeline and expose failure

Dimension	Question	Examples of institutionally actionable evidence
Detectability	Can the affected person or reviewer know that an AI-assisted consequential action occurred?	Notice sent or scheduled, visible case-note reference, disclosure of agent involvement where required
Traceability	Can the action be traced to tools, records, policy, model/agent configuration, and responsible actors?	Tool name, action ID, evidence handles, policy IDs, actor stack, timestamp
Contestability	Can the affected person challenge the action through a meaningful process?	Appeal route, review owner, deadline, evidence package, non-retaliation or access provisions
Reversibility	Can the action be undone, corrected, or paused while contested?	Rollback endpoint, restoration procedure, reversible-state flag, irreversible-action justification
Remedy	Can the person obtain repair beyond reversal?	Compensation, service restoration, apology, corrected record, priority restoration, fee waiver
Recurrence prevention	Does the institution learn without hiding evidence?	Incident tag, policy update path, retraining caveat, audit trigger, monitoring rule
Burden allocation	Who must prove what?	Institution provides evidence by default, and the affected person is not required to infer hidden records
Value-chain assignability	Can responsibility be allocated across model provider, agent builder, tool provider, deployer, and supervisor?	Actor stack and ownership of model, tool, data source, deploying unit, and human approval

Table 2: Redressability dimensions used in diagnostic scoring and blinded validation.

Scenario field	Purpose
Agent role and tools	Defines delegated authority and possible side effects.
Policy snippets	Specifies evidence thresholds, notice, approval, and appeal rules.
Records	Provides support, ambiguity, or contradiction for action.
Affected party	Identifies who needs notice, appeal, or remedy.
Unsafe targets	Hidden labels for diagnostic scoring and validation.
Expected safe action	Used only for evaluation, never shown to the agent.
Contestation prompt	Supports judgment of whether a later challenge would be possible.

Table 3: Core scenario fields in Agent RedressBench.

modes. GPT-4.1 was evaluated on the 40-scenario v0.3 seed set and the 24-scenario v0.4 stress split; Ollama/Llama 3.1 8B was evaluated on the v0.3 seed set. Each model-condition-scenario tuple produced one structured trajectory, yielding 160 GPT-4.1 seed trajectories, 96 GPT-4.1 stress trajectories, and 160 Llama trajectories. The automatic scorer measured parse success, backend errors, trace presence, diagnostic redressability, consequential-action status, and provisional unsafe-action labels. These automatic labels are high-recall diagnostic flags rather than validated safety judgments, particularly because the stress split includes substantively valid consequential actions whose principal defect may be missing redress evidence.

Implementation details. The OpenAI runs used the API model identifier `gpt-4.1` on 10 May 2026; the local runs used the Ollama model tag `llama3.1:8b` on 11 May 2026.

```

1  if consequential(call):
2      req = predicates(policy, call)
3      if not holds(req, trace):
4          block(call)
5          store_block(call, trace)
6          escalate(policy.owner)
7      else:
8          execute(call)
9          store(trace)
10 else:
11     execute(call)

```

Figure 1: Pseudocode sketch of runtime-gated redress. A consequential tool call proceeds only when the policy-required redress predicates are satisfied; otherwise, the call is blocked, logged, and escalated. The gate is not evaluated as an experimental condition in this study.

Exact scenario files, condition prompts, output contracts, model identifiers, timestamps, structured JSONL trajectories, and scorer outputs are included in the artifact. Each cell was run once. No malformed GPT-4.1 or Llama trajectory remained in the reported sets, and no reported call ended with a backend error. Because the run records do not preserve every provider-side serving detail, the results should be interpreted as reproducible at the artifact and model-identifier level rather than as evidence tied to an immutable hosted-model snapshot.

Second, a blinded single-validator assessment evaluates construct validity on all 96 GPT-4.1 stress trajectories. The validator received sanitized case files without condition labels, automatic scores, expected-action fields, run identifiers, or the private condition mapping, and assigned a safety judgment plus eight ordinal redressability scores. Because the assessment uses one validator and no independent adjudication, it is treated as construct-validation evidence rather than

Condition	Safety instruction	Trace required	Escalation required	Tool gate	Interpretation
Baseline	✗	✗	✗	✗	Measures ordinary task-completion behavior under the supplied institutional context.
Guardrailed	✓	✗	Partial	✗	Tests whether prompt-level safety rules reduce risky action without redress artifacts.
Trace	Partial	✓	✗	✗	Tests whether action-level redress artifacts can be generated during consequential action.
Trace + escalation	✓	✓	✓	✗	Tests whether redress traces plus conservative escalation reduce unredressable action.
Proposed runtime-gated design	✓	✓	Conditional	✓	Blocks side-effectful tools unless policy-defined redress predicates hold; not evaluated here.

Table 4: Evaluated experimental conditions and proposed runtime-gated design. The first four rows are evaluated; the final row is a design implication and is not included in the reported experiments.

Dimension	Environment supplies (held fixed)	Scored trajectory must contribute
Detectability	Notification tools and disclosure rules	Send or schedule notice and disclose agent role
Traceability	Trace schema and action IDs	Complete, accurate action-level trace
Contestability	Appeal route and review owner in policy	Cite route, owner, and deadline with evidence package
Reversibility	Rollback endpoints where contemplated	Use or record rollback path, or justify irreversibility
Remedy	Remedy options permitted by policy	Offer the applicable remedy options
Recurrence prevention	Incident and audit channels	File incident tag and propose prevention step
Burden allocation	Default burden rules in policy	Keep evidentiary burden on the institution
Value-chain assignability	Actor-stack fields in schema	Name model, tool, employer, and approver

Table 5: Locus of responsibility for each dimension. Scenarios supply the institutional affordances (middle column), so dimension scores measure the trajectory’s contribution (right column), not the institution’s maturity.

inter-rater-confirmed ground truth.

Analysis. Scenarios are matched across conditions. Automatic diagnostic outcomes are reported descriptively as rates and paired directional changes because the proxy labels are not ground truth. For the single-validator redressability ratings, we report paired mean differences, percentile bootstrap 95% confidence intervals, and two-sided Wilcoxon signed-rank tests. These inferential comparisons are exploratory because they reflect one validator rather than an adjudicated multi-rater study.

Predicate	Gate behavior
Approval	Block adverse action unless approval is obtained or a documented emergency exception applies.
Notice	Require affected-party notice to be sent or scheduled before or immediately after action.
Evidence	Require evidence handles and policy basis for the action.
Appeal	Require a usable review route with owner and deadline.
Rollback	Require a rollback path or explicit irreversible-action justification.
Remedy	Require at least one remedy option when action can impose cost or loss.
Privacy	Reject traces that expose raw sensitive content when handles suffice.

Table 6: Policy predicates in the proposed runtime-gated redress design. The gate is not evaluated as an experimental condition in this study.

Diagnostic Results

GPT-4.1 Seed and Stress Results

GPT-4.1 completed all v0.3 seed and v0.4 stress trajectories with 100% parse success and no backend errors. On the 40-scenario seed set it produced no automatic adverse-side-effect harms in any condition, revealing a ceiling effect: the seed set is useful for testing trace generation but too easy for strong-model harm-prevention claims. Trace conditions nevertheless produced nonempty traces in all cases and mean diagnostic redressability scores of 20.0/24 and 20.9/24 for trace and trace + escalation, respectively. Baseline and guardrailed conditions scored 0 because their contracts did not require redress traces.

The 24-scenario stress split partially addressed this ceiling effect. Baseline and guardrailed conditions took consequential actions without redress artifacts in 10/24 and 12/24 cases,

respectively. Trace and trace + escalation produced no unredressable consequential-action cases under the automatic scorer. Table 7 summarizes the stress split.

The stress split also exposes a tradeoff. Trace + escalation produced fewer autonomous valid consequential actions than trace-only (5/24 versus 11/24) and more safe escalations. This may represent appropriate conservatism in high-impact workflows or excessive deferral. The blinded validation therefore assesses safety and redressability separately rather than treating escalation as inherently desirable.

Llama 3.1 8B Seed Results

The local-model run is more revealing for safety variation. Ollama/Llama 3.1 8B completed all 160 seed trajectories with 100% parse success and no backend errors. The guardrailed condition reduced automatic unsafe-action labels from 28/40 in baseline to 10/40. Trace-only produced redress artifacts in all cases but still received unsafe-action labels in 26/40 cases. Trace + escalation reduced this to 16/40, but remained less safe under the automatic scorer than guardrails alone. Table 8 summarizes the result.

This result is a useful negative finding. Redress traces are not safety mechanisms by themselves. In several trace-condition trajectories, the model filled the approval field with *required: true* and *status: not obtained*, while still executing an adverse tool call. We call this a **procedural-trace failure**: the agent documents the procedural defect but does not let the procedure constrain action. The pattern is consistent with broader evidence that LM agents can articulate risk knowledge without acting on it (Tang et al. 2025). A runtime gate is designed to prevent exactly this failure.

Paired diagnostic comparisons make the pattern sharper. Guardrails reduced automatic unsafe-action labels relative to baseline by 45 percentage points: 18 scenarios changed from unsafe to safe and no scenario changed from safe to unsafe. Trace-only was not meaningfully different from baseline on the unsafe-action label. Trace + escalation reduced unsafe labels relative to trace-only by 25 percentage points, but it still produced more automatic unsafe labels than guardrails alone. In other words, prompt-level redress improved accountability artifacts, while guardrails improved immediate action safety. Neither alone achieved both goals.

Domain results reinforce the need to separate proxy labels from case-level procedural judgment. Customer-support scenarios were hardest for Llama 3.1 8B (unsafe-action rates 0.90, 0.60, 0.80, and 0.80 across the four conditions), while healthcare-admin scenarios were more responsive to guardrails and escalation (0.60, 0.10, 0.60, and 0.10). Redress requirements may therefore need domain-specific thresholds rather than a single universal policy.

Qualitative Stress Cases

Full-trajectory inspection of GPT-4.1 stress cases clarifies why redressability is not equivalent to refusing action. In STR-WOR-002, the baseline and guardrailed agents restricted badge access based on confirmed security and termination records. The action may be substantively valid, but the output did not include a top-level redress trace, so the diagnostic score was 0/24. In the trace condition, the agent

took the supported access action while preserving evidence handles, notice, appeal route, actor stack, and rollback fields. In STR-WOR-004, baseline created a temporary payroll-hold record without a structured trace, while trace + escalation preserved the evidence package and shifted final authority to human HR review.

These cases, together with the failure modes cataloged in Table 9, motivate case-level validation that asks not merely whether the outcome was substantively acceptable, but whether an affected person would have a usable institutional path to challenge, reverse, and repair it.

Single-Validator Blinded Validation

A single blinded validator assessed all 96 GPT-4.1 v0.4 stress trajectories for safety and the eight redressability dimensions. The validator did not receive condition labels, automatic scores, hidden expected-action fields, run identifiers, or the private mapping from blinded case IDs to conditions. The completed file passed structural checks for row coverage, unique case IDs, allowed labels, valid 0–3 dimension scores, and absence of prohibited blinding terms. Because the assessment includes one validator and no independent adjudication, the ratings are used as construct-validation evidence rather than adjudicated ground truth.

The assessment did not substantiate a strong GPT-4.1 harm-prevention claim: 93 trajectories were judged safe, three ambiguous, and none unsafe (Table 10). It instead supports the paper’s central distinction between substantive safety and procedural redressability. Trace requirements increased mean redressability by 14.83 points over baseline (95% bootstrap CI [13.79, 15.92], Wilcoxon $p < .001$), and trace plus escalation increased it by 14.96 points (95% CI [14.04, 15.96], $p < .001$). Guardrails alone did not materially improve redressability over baseline (mean difference 0.21, 95% CI [−0.54, 1.08], $p = .863$). Trace plus escalation likewise did not improve redressability over trace alone (mean difference 0.13, 95% CI [−0.46, 0.75], $p = .937$), although it changed the action policy: 19/24 trajectories were safe escalations compared with 8/24 under trace-only.

The three ambiguous cases involved a refusal despite supporting evidence, an access-sensitive communication sent to a recently updated contact without additional verification, and a refund denial lacking required notice. They illustrate why safety and redressability must be scored separately: a procedural omission may reduce redressability in one setting but constitute a safety-relevant violation in another. Multi-rater assessment would improve reliability estimates, but the present evidence is sufficient to test whether the dimensions distinguish trace from non-trace trajectories.

Discussion

The results support four design and evaluation principles for redressable agents.

Action-level provenance should be mandatory for consequential tools. Traceability should be attached to the tool call rather than reconstructed from a later natural-language summary. A reviewer should be able to recover the action ID, evidence handles, policy basis, approval state, and actor

Condition	N	Unsafe	Conseq.	Valid conseq.	Unredress. conseq.	Trace	Redress mean
Baseline	24	0.083	0.417	0.333	0.417	0.00	0.00
Guardrailed	24	0.042	0.500	0.458	0.500	0.00	0.00
Trace	24	0.000	0.458	0.458	0.000	1.00	20.875
Trace + escalation	24	0.000	0.208	0.208	0.000	1.00	21.208

Table 7: GPT-4.1 v0.4 stress-split diagnostic summary. “Unredress. conseq.” denotes a consequential action with diagnostic redressability score 0. Unsafe labels are automatic diagnostic classifications, not adjudicated safety judgments.

Condition	N	Unsafe count	Unsafe rate	Proxy rate	Trace	Redress mean
Baseline	40	28	0.700	0.950	0.00	0.00
Guardrailed	40	10	0.250	0.425	0.00	0.00
Trace	40	26	0.650	0.750	1.00	21.10
Trace + escalation	40	16	0.400	0.500	1.00	21.13

Table 8: Ollama/Llama 3.1 8B v0.3 seed-run diagnostic summary. Redressability has maximum 24. Unsafe labels are automatic diagnostic classifications, not adjudicated safety judgments.

stack from the stored trajectory. This requirement is compatible with privacy because traces can use handles, hashes, and record references instead of embedding sensitive content.

Redress predicates should govern execution. Prompt instructions are insufficient when the model itself decides whether its procedural commitments are binding. The proposed runtime gate is a design implication rather than an evaluated treatment: it specifies how a deployed system could reject a tool call when approval, notice, evidence, appeal, rollback, remedy, or privacy requirements are missing. In such an architecture, redressability would become a safety invariant over side-effectful tool use rather than a documentation practice. The present results motivate this design through observed procedural-trace failures but do not estimate its treatment effect.

Redress should preserve valid action. A redressable agent should not simply escalate every high-impact case. Over-escalation can delay services, create administrative burden, and become a form of denial. The appropriate target is conditional autonomy: complete valid consequential actions when substantive and procedural predicates are satisfied, and escalate when they are not. The GPT-4.1 comparison between trace and trace plus escalation illustrates this tradeoff: escalation frequency changed sharply while validated redressability remained nearly constant.

Redress quality should be audited independently of model quality. A highly capable model can produce low-redress trajectories when the system contract does not require review artifacts. Conversely, a weaker model can produce detailed artifacts while ignoring their procedural implications. Redressability therefore belongs at the system level, where prompts, output contracts, tool runtimes, records, policies, and organizational workflows jointly determine whether repair is possible.

From benchmark score to deployment decision

A high redressability score should not be interpreted as permission to deploy. It indicates that one necessary condition

for accountable deployment has been tested. Deployment would additionally require domain-specific legal analysis, privacy review, organizational capacity to process appeals, monitoring for disparate burden, and evidence that affected people can access the promised remedy. A benchmark can evaluate whether a trajectory preserves procedural affordances, but it cannot guarantee that a real institution will honor them fairly.

Redress is an institutional performance

A trajectory can carry procedural hooks: notice sent, evidence preserved, appeal route named, rollback path recorded, remedy options offered, and responsible actors identified. Redress itself is completed by the institution in which the agent is embedded. Repair can require reversal, record correction, and restoration of rights, as well as acknowledgment and apology, the satisfaction component of remedy (UN General Assembly 2005), and consequences for responsible actors (Bovens 2007; EU AI Act 2024). A trajectory cannot perform all of these institutional acts, but it can make them possible. Effective grievance mechanisms are legitimate, accessible, predictable, equitable, and transparent (United Nations 2011); a redress trace is the evidentiary input such a mechanism needs when the acting system is an agent. Trace infrastructure without institutional capacity is paperwork, whereas institutional capacity without trace infrastructure is blind.

Artifact Availability

The repository contains scenario files and schemas, condition prompts and output contracts, diagnostic scoring and run-audit code, sanitized validation cases, completed blinded ratings, aggregate summaries, and validation utilities. Public materials are separated from private condition mappings and validator-identifying information. The artifact supports reproduction of the reported tables and later multi-rater validation without revealing hidden labels.

Failure mode	Diagnostic signature	Design implication
Unredressable valid action	Supported consequential action with no usable top-level trace	Substantive validity does not guarantee contestability or repair.
Procedural-trace failure	Trace records a missing approval while the adverse tool call still executes	Procedural evidence must constrain execution, not merely document it.
Over-escalation	Supported actions are routed to review rather than completed	Conservatism can create delay, workload, and denial-by-process.
Field-level trace inflation	Appeal or remedy terms appear in tool arguments but not in a structured artifact	Keyword presence is not equivalent to an actionable process.
Value-chain opacity	Tool is identified but accountable model, deployer, reviewer, or policy owner is absent	Remedy requires responsibility to be assignable across the value chain.

Table 9: Failure modes identified through diagnostic trajectory analysis and examined in the blinded validation.

Condition	N	Safe	Ambig.	Unsafe	<i>R</i> mean
Baseline	24	22	2	0	4.38
Guardrailed	24	23	1	0	4.58
Trace	24	24	0	0	19.21
Trace + escalation	24	24	0	0	19.33

Table 10: Single-validator blinded assessment of the 96 GPT-4.1 stress trajectories. Redressability *R* has maximum 24. The ratings are not multi-rater or adjudicated and are interpreted as construct-validation evidence.

Ethics and Release

All scenarios are synthetic composites and contain no private records or data from represented case subjects. The single-validator assessment collected no demographic or other sensitive information and is presented as artifact and construct validation rather than an adjudicated human-subjects study. Any subsequent annotation study that recruits students, employees, contractors, or domain experts should obtain the locally required ethics determination and address voluntariness, supervision, compensation, and data minimization.

Redress traces can themselves create risks: audit trails may become surveillance infrastructure, expose sensitive information, or provide procedural cover for harmful decisions. The schema therefore favors evidence handles over raw records, requires a privacy-minimization field, and treats trace completeness as insufficient unless the surrounding institution actually supports notice, review, rollback, and remedy.

Limitations

The benchmark uses synthetic scenarios, which improves experimental control and privacy but underrepresents real organizational ambiguity, data-quality failures, legal variation, and power asymmetries. Its scenarios assume procedurally mature institutions with functioning notice, appeal, review, and rollback affordances; the study does not test whether real organizations honor those mechanisms. The automatic scorer is diagnostic and cannot determine whether a remedy is meaningful or an action substantively justified. GPT-4.1 exhibited a ceiling effect on the seed set and limited automatic harm variation on the stress split, whereas Llama 3.1 8B produced richer failure variation but may not generalize

to other local models. Redressability is also partly normative, and domains may legitimately weight reversal, remedy, and escalation differently.

Threats to validity. Prompt differences may affect behavior beyond the intended condition manipulation, and redress fields may prime caution even without enforcement. The automatic scorer may under-credit redress information buried in tool arguments or over-credit vague structured fields. The single-validator assessment establishes construct separation but does not provide inter-rater reliability or adjudicated labels. Results are based on one trajectory for each model, condition, and scenario combination and on model identifiers that may not uniquely determine future hosted behavior.

Empirical scope. The reported experiments evaluate baseline, guardrailed, trace, and trace-plus-escalation conditions. REDRESS GATE is a design implication motivated by the observed procedural-trace failure, not a measured intervention, and the paper makes no claim about its treatment effect. Likewise, the single-validator ratings should not be interpreted as population-level estimates of human judgment. Multi-rater validation, domain-expert assessment, and direct evaluation of runtime gating remain necessary to establish reliability and deployment relevance.

Conclusion

Tool-using AI agents create institutional events, not only textual outputs. Agent safety should therefore ask not only whether harmful actions are prevented, but whether consequential actions can be discovered, attributed, contested, reversed, and remedied. This work operationalizes redressability, a long-standing legal and institutional concept, as an evaluable property of tool-using agents; introduces a benchmark and action-level trace schema; and shows that guardrails, trace generation, and escalation address distinct failure modes. The blinded validation separates trace from non-trace trajectories in redressability while finding little evidence of unsafe GPT-4.1 action, confirming that substantive safety and post-action repair are non-equivalent. Weaker-model procedural failures motivate runtime-gated redress as a design implication.

References

- Andriushchenko, M.; Souly, A.; Dziemian, M.; Duenas, D.; Lin, M.; Wang, J.; Hendrycks, D.; Zou, A.; Kolter, J. Z.; Fredrikson, M.; Gal, Y.; and Davies, X. 2025. AgentHarm: A Benchmark for Measuring Harmfulness of LLM Agents. In *International Conference on Learning Representations*.
- Bovens, M. 2007. Analysing and Assessing Accountability: A Conceptual Framework. *European Law Journal*, 13(4): 447–468. doi:10.1111/j.1468-0386.2007.00378.x.
- Chan, A.; Salganik, R.; Markelius, A.; Pang, C.; Rajkumar, N.; Krashennnikov, D.; Langosco, L.; He, Z.; Duan, Y.; Carroll, M.; Lin, M.; Mayhew, A.; Collins, K.; Molamohammadi, M.; Burden, J.; Zhao, W.; Rismani, S.; Voudouris, K.; Bhatt, U.; Weller, A.; Krueger, D.; and Maharaj, T. 2023. Harms from Increasingly Agentic Algorithmic Systems. In *Proceedings of the 2023 ACM Conference on Fairness, Accountability, and Transparency*, 651–666. doi:10.1145/3593013.3594033.
- Chan, A.; Ezell, C.; Kaufmann, M.; Wei, K.; Hammond, L.; Bradley, H.; Bluemke, E.; Rajkumar, N.; Krueger, D.; Kolt, N.; Heim, L.; and Anderljung, M. 2024. Visibility into AI Agents. In *Proceedings of the 2024 ACM Conference on Fairness, Accountability, and Transparency*, 958–973. doi:10.1145/3630106.3658948.
- Chappidi, S.; Cobbe, J.; Norval, C.; Mazumder, A.; and Singh, J. 2025. Accountability Capture: How Record-Keeping to Support AI Transparency and Accountability (Re)shapes Algorithmic Oversight. *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 8(1): 554–566. doi:10.1609/aies.v8i1.36570.
- Citron, D. K. 2008. Technological Due Process. *Washington University Law Review*, 85(6): 1249–1313.
- Citron, D. K.; and Pasquale, F. 2014. The Scored Society: Due Process for Automated Predictions. *Washington Law Review*, 89(1): 1–34.
- Cobbe, J.; Lee, M. S. A.; and Singh, J. 2021. Reviewable Automated Decision-Making: A Framework for Accountable Algorithmic Systems. In *Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency*, 598–609. doi:10.1145/3442188.3445921.
- Cooper, A. F.; Moss, E.; Laufer, B.; and Nissenbaum, H. 2022. Accountability in an Algorithmic Society: Relationality, Responsibility, and Robustness in Machine Learning. In *Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency*, 864–876. doi:10.1145/3531146.3533150.
- European Parliament and Council of the European Union. 2024. Regulation (EU) 2024/1689 of the European Parliament and of the Council of 13 June 2024 Laying Down Harmonised Rules on Artificial Intelligence and Amending Regulations (EC) No 300/2008, (EU) No 167/2013, (EU) No 168/2013, (EU) 2018/858, (EU) 2018/1139 and (EU) 2019/2144 and Directives 2014/90/EU, (EU) 2016/797 and (EU) 2020/1828 (Artificial Intelligence Act). *Official Journal of the European Union*, OJ L 2024/1689, 12 July 2024.
- Ezell, C.; Roberts-Gaal, X.; and Chan, A. 2025. Incident Analysis for AI Agents. *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 8(1): 865–878. doi:10.1609/aies.v8i1.36596.
- Hopkins, A.; Cen, S. H.; Struckman, I.; Ilyas, A.; Videgaray, L.; and Madry, A. 2025. AI Supply Chains: An Emerging Ecosystem of AI Actors, Products, and Services. *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 8(2): 1266–1277. doi:10.1609/aies.v8i2.36628.
- Kaminski, M. E.; and Urban, J. M. 2021. The Right to Contest AI. *Columbia Law Review*, 121(7): 1957–2048.
- Karimi, A.-H.; Barthe, G.; Schölkopf, B.; and Valera, I. 2022. A Survey of Algorithmic Recourse: Contrastive Explanations and Consequential Recommendations. *ACM Computing Surveys*, 55(5), Article 95: 1–29. doi:10.1145/3527848.
- Kroll, J. A. 2018. The Fallacy of Inscrutability. *Philosophical Transactions of the Royal Society A*, 376(2133): 20180084. doi:10.1098/rsta.2018.0084.
- Lind, E. A.; and Tyler, T. R. 1988. *The Social Psychology of Procedural Justice*. New York: Plenum Press.
- Lujan v. Defenders of Wildlife. 1992. 504 U.S. 555. Supreme Court of the United States.
- Lyons, H.; Velloso, E.; and Miller, T. 2021. Conceptualising Contestability: Perspectives on Contesting Algorithmic Decisions. *Proceedings of the ACM on Human-Computer Interaction*, 5(CSCW1), Article 106: 1–25. doi:10.1145/3449180.
- McGregor, S. 2021. Preventing Repeated Real World AI Failures by Cataloging Incidents: The AI Incident Database. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(17): 15458–15463. doi:10.1609/aaai.v35i17.17817.
- Mökander, J.; Schuett, J.; Kirk, H. R.; and Floridi, L. 2024. Auditing Large Language Models: A Three-Layered Approach. *AI and Ethics*, 4: 1085–1115. doi:10.1007/s43681-023-00289-2.
- Novelli, C.; Taddeo, M.; and Floridi, L. 2024. Accountability in Artificial Intelligence: What It Is and How It Works. *AI & Society*, 39(4): 1871–1882. doi:10.1007/s00146-023-01635-y.
- Raji, I. D.; Smart, A.; White, R. N.; Mitchell, M.; Gebru, T.; Hutchinson, B.; Smith-Loud, J.; Theron, D.; and Barnes, P. 2020. Closing the AI Accountability Gap: Defining an End-to-End Framework for Internal Algorithmic Auditing. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 33–44. doi:10.1145/3351095.3372873.
- Rao, P. S. B.; Šćepanović, S.; Jayagopi, D. B.; Cherubini, M.; and Quercia, D. 2025. The AI Model Risk Catalog: What Developers and Researchers Miss About Real-World AI Harms. *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 8(3): 2150–2163. doi:10.1609/aies.v8i3.36702.
- Riedl, M. O.; and Desai, D. R. 2025. AI Agents and the Law. *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 8(3): 2189–2198. doi:10.1609/aies.v8i3.36705.
- Rismani, S.; Shelby, R.; Davis, L.; Rostamzadeh, N.; and Moon, A. 2025. Measuring What Matters: Connecting AI Ethics Evaluations to System Attributes, Hazards, and Harms. *Proceedings of the AAAI/ACM Conference on AI, Ethics, and Society*, 8(3): 2199–2213. doi:10.1609/aies.v8i3.36706.

- Ruan, Y.; Dong, H.; Wang, A.; Pitis, S.; Zhou, Y.; Ba, J.; Dubois, Y.; Maddison, C. J.; and Hashimoto, T. 2024. Identifying the Risks of LM Agents with an LM-Emulated Sandbox. In *International Conference on Learning Representations*.
- Santoni de Sio, F.; and Mecacci, G. 2021. Four Responsibility Gaps with Artificial Intelligence: Why They Matter and How to Address Them. *Philosophy & Technology*, 34(4): 1057–1084. doi:10.1007/s13347-021-00450-x.
- Selbst, A. D.; Boyd, D.; Friedler, S. A.; Venkatasubramanian, S.; and Vertesi, J. 2019. Fairness and Abstraction in Sociotechnical Systems. In *Proceedings of the Conference on Fairness, Accountability, and Transparency*, 59–68. doi:10.1145/3287560.3287598.
- Shapira, A.; Gandhi, P. A.; Habler, I.; and Shabtai, A. 2026. Mind the Web: The Security of Web Use Agents. In *Proceedings of the 21st ACM Asia Conference on Computer and Communications Security*, 835–851. doi:10.1145/3779208.3805968.
- Shavit, Y.; Agarwal, S.; Brundage, M.; Adler, S.; O’Keefe, C.; Campbell, R.; Lee, T.; Mishkin, P.; Eloundou, T.; Hickey, A.; Slama, K.; Ahmad, L.; McMillan, P.; Beutel, A.; Passos, A.; and Robinson, D. G. 2023. Practices for Governing Agentic AI Systems. Technical report, OpenAI.
- Tabassi, E. 2023. Artificial Intelligence Risk Management Framework (AI RMF 1.0). NIST AI 100-1. Gaithersburg, MD: National Institute of Standards and Technology. doi:10.6028/NIST.AI.100-1.
- Tang, Y.; Li, T.; Li, E.; Maddison, C. J.; Dong, H.; and Ruan, Y. 2025. LM Agents May Fail to Act on Their Own Risk Knowledge. *arXiv preprint arXiv:2508.13465*.
- Tur, A. D.; Meade, N.; Lù, X. H.; Zambrano, A.; Patel, A.; Durmus, E.; Gella, S.; Stanczak, K.; and Reddy, S. 2025. SafeArena: Evaluating the Safety of Autonomous Web Agents. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, 60404–60441.
- Tyler, T. R. 2006. *Why People Obey the Law*. Princeton, NJ: Princeton University Press. With a new afterword by the author.
- United Nations. 2011. Guiding Principles on Business and Human Rights: Implementing the United Nations “Protect, Respect and Remedy” Framework. HR/PUB/11/4, Office of the United Nations High Commissioner for Human Rights.
- United Nations General Assembly. 2005. Basic Principles and Guidelines on the Right to a Remedy and Reparation for Victims of Gross Violations of International Human Rights Law and Serious Violations of International Humanitarian Law. Resolution A/RES/60/147.
- Vijayvargiya, S.; Soni, A. B.; Zhou, X.; Wang, Z. Z.; Dziri, N.; Neubig, G.; and Sap, M. 2026. OpenAgentSafety: A Comprehensive Framework for Evaluating Real-World AI Agent Safety. In *International Conference on Learning Representations*.
- Wachter, S.; Mittelstadt, B.; and Russell, C. 2018. Counterfactual Explanations Without Opening the Black Box: Automated Decisions and the GDPR. *Harvard Journal of Law & Technology*, 31(2): 841–887.
- Wieringa, M. 2020. What to Account for When Accounting for Algorithms: A Systematic Literature Review on Algorithmic Accountability. In *Proceedings of the 2020 Conference on Fairness, Accountability, and Transparency*, 1–18. doi:10.1145/3351095.3372833.
- Yao, S.; Shinn, N.; Razavi, P.; and Narasimhan, K. 2025. τ -bench: A Benchmark for Tool-Agent-User Interaction in Real-World Domains. In *International Conference on Learning Representations*.