A Verification-Gated Architecture for AI-Assisted Research Production in the Humanities

Working paper, draft v27 — version timestamp 2026-08-24T13:20:06-04:00
HYDER HUSAIN ARASTU, University of South Florida, USA; AAKASH SHAHANI, Independent, USA; TAHER ALI, Independent, USA

This paper presents a verification-gated architecture for AI-assisted research production in the humanities: a systems and infrastructure contribution, not an evaluative one in the human-outcome sense, over a closed, pre-manifested corpus. It is built amid – not against – the risk that unchecked AI assistance erodes critical thinking, rather than the certainty that it does. That risk has moved from shared worry to peer-reviewed finding, and the evidence locates the harm in one avoidable design choice – answer delivery without a checkable step – not in generative AI. A mechanical, script-run check settles whether a quotation resolves against that corpus; a rule-based join settles whether a claim inherits its status; neither settles paraphrase fidelity or source trust, decisions reserved for a human or an independent reviewing agent at every result, not only failures. We demonstrate the architecture by applying it to itself, in a meta-project run by two parallel moderating agents – one conducting this research, the other building a reference implementation of the design pattern under study. Because both lines instance one unified research method, that implementation carries no separate evaluation here: it is presented only as the architecture's own worked instance, not as a second, reader-facing system under test. Both agents worked under one shared architecture across cycles of drafting, adversarial review, and re-verification against historical, committed snapshots. No empirical result about a reader's thinking is claimed; the evidence is the meta-project's own audit trail, and a narrower, single-project methodological claim rests on that same evidence. We differentiate the architecture from comparable AI-assisted-research systems by domain, corpus role, and terminal check.

CCS Concepts: • **Human-centered computing** → **HCI theory, concepts and models**; *Interaction paradigms*; • **Applied computing** → *Arts and humanities*.

Additional Key Words and Phrases: critical thinking, verification, epistemic erosion, AI-assisted scholarship, research production

Working paper, draft v27 — version timestamp 2026-08-24T13:20:06-04:00.
Public archival copy; byline restored.

1 Introduction

The fear that new tools erode a reader's own thinking has, once again, moved from shared unease into the peer-reviewed record. A mixed-methods study of 666 participants finds frequent AI-tool use correlating with lower critical-thinking scores, an association its own design keeps correlational rather than causal [Gerlich 2025]. An EEG study of essay writing with a chatbot reports weaker neural and linguistic engagement [Kosmyna et al. 2025], a finding a subsequent methodological commentary presses – sample size, EEG-analysis reproducibility, reporting completeness – without overturning the underlying concern [Stankovic et al. 2026].

None of this is new in kind: the worry that a technology will do a mind's work for it and leave the mind weaker recurs across the history of writing technologies; large-language models are only its latest occasion.

What the recent literature adds is design-contingency. In a large field study across high-school mathematics classrooms, the same language model raised or lowered post-test performance depending only on its guardrails: an unguarded chat interface let students copy answers, leaving later unassisted performance worse than a no-AI control, while a guardrail-equipped version produced no such harm [Bastani et al. 2025]. This is a K-12 mathematics finding; this paper extends it to AI-assisted research production in the humanities by domain analogy, not direct evidence, flagged as such from its first use here. The erosion the literature documents is a property of one common design choice – answer delivery without a checkable step – not of generative AI as such, surveyed next. This paper names that design-contingent property *epistemic erosion*: the hypothesis that an AI-assisted workflow degrades a reader's own critical thinking when it substitutes answer delivery for a checkable step, not a demonstrated general effect of AI assistance [Bastani et al. 2025; Gerlich 2025].

Philosophy of cognitive science already poses the same problem as an open design question. Closing an argument about which digital resources count as part of a person's own cognitive apparatus, Andy Clark poses, as a question for future research rather than a settled requirement, how the designers and users of new tools might exercise due epistemic caution [Clark 2015]. Clark draws the historical line himself: Plato's *Phaedrus* already gives "a clear statement of the fear that new-fangled inventions such as reading and writing will have catastrophic effects on human memory," and on the same page carries that fear forward to the present case [Clark 2025, p. 1]. The question this paper takes up is Clark's: not whether such a tool can erode a reader's thinking – the literature above already answers that, for one class of designs – but how its design might instead make that thinking checkable.

We present a verification-gated architecture for AI-assisted research production in the humanities: a gated pipeline holding a closed, pre-manifested corpus, exposing AI-offloaded quotation and citation-linkage operations to three explicitly typed checks – a mechanical, script-run verbatim-substring match, a rule-based claim-inheritance join, and a severity-graded human-and-agent review. A human-ratification point is reserved for every result, not only failures, and each check is re-verified against historical, committed snapshots rather than trusted from a single run. The first check settles whether a quotation resolves; the second, only whether a claim inherits that status, never whether its paraphrase is a faithful entailment; the third is explicitly interpretive, not mechanical. This is a design and demonstration claim, not an empirical one.

We demonstrate the architecture by applying it to itself. Two parallel moderating agents ran this project: one conducted the research and drafted this paper, the other served as the parallel build agent, constructing a reference implementation of the design pattern under study. Because both lines instance one unified research method (§5), that implementation carries no separate evaluation of its own: it is checked only by the research line's read-only verification against its own source, drawn on below to show how the architecture aids research, not tested as a second, reader-facing system. This is a common-mode limitation stated plainly in §8, not withheld until then. The project is accordingly a meta-project for the architecture it describes, carried out across iterative cycles of drafting, adversarial review, and re-verification. What this design and demonstration claim does not license is stated in one place next (§3).

This paper makes four contributions. We **reframe** the critical-thinking-erosion debate, arguing from causal and design-contingent evidence that erosion is a structural design risk, not a demonstrated property of

AI use as such (§1–§2). We **derive** numbered design desiderata from the appropriate-reliance, cognitive-load, and cognitive-offloading literature and **specify** a verification-gated architecture that satisfies them, typed to the three checks above (§4–§6). That architecture's record-keeping tooling includes a selective, hash-triggered invalidation engine, backstopped by the full resweep (§6). We **report** a worked feasibility case: applying the architecture to its production surfaced the documented failure modes that motivate it – hallucinated citation, silent overclaim, false-negative extraction artifacts, unaccountable delegation – each caught by the architecture's mechanisms on this project, not established as a general property (§7). That evidence is the research line's process, not a reader's use of the reference implementation, whose correspondence to this architecture remains a design thesis, not itself demonstrated (§3–§4). We **differentiate** this architecture from agentic-discovery systems, claim-verification benchmarks, and provenance-tracing architectures on domain, corpus role, and terminal check, detailed in §2.

2 Related Work

2.1 Erosion and design-contingent enhancement

A CHI-native literature treats generative AI's metacognitive costs as a property of design, not the tool itself. Tankelevitch et al. locate metacognitive demand at framing a prompt, calibrating trust, and deciding whether to apply the tool at all [Tankelevitch et al. 2024]. Passi and Vorvoreanu name the failure that last demand produces, overreliance on an incorrect recommendation, and its most-cited remedy, cognitive forcing functions, while warning an appeal to human oversight "can provide a false sense of security" where nothing enforces it [Passi and Vorvoreanu 2022]. Buçinca et al. supply the founding evidence and its cost: forcing functions reduced overreliance, and participants rated the best-working designs least favorably [Buçinca et al. 2021]. Vendrell and Johnston carry the design-contingency claim into higher education, arguing LLM effects are "contingent rather than fixed" [Vendrell and Johnston 2026]; Drosos et al. find AI-generated provocations induced critical thinking, shaped by task urgency and user expertise [Drosos et al. 2025]. Tian, Amin, and Yin's AI-Assisted Critical Thinking framework, prompting reflection on a decision-maker's own reasoning rather than delivering an answer outright, reduced AI over-reliance in a house-price-prediction case study, at the cost of higher cognitive load [Tian et al. 2026].

Lee et al.'s survey of knowledge workers is the sharpest hook for our argument: confidence in generative AI predicts less enacted critical thinking, while what remains shifts toward verification and stewardship [Lee et al. 2025]. Two studies bound how far that shift can be designed for. Zhi, Kumar, and Lee find a temporal reversal: early access helps under time pressure and hurts with time to spare [Zhi et al. 2026a]. Zhi, Long, So, and Lee find dose is what matters instead: one AI interpretation aids comprehension, while heavy reliance trades it for foreclosure [Zhi et al. 2026b]. Bae et al.'s thirteen-participant study adds the reader as a further bound: the same unscaffolded assistance raised critical-thinking scores for experienced researchers and lowered them, with high variance, for novices, a result its authors call "indicative rather than definitive" [Bae et al. 2026]. Bo, Wan, and Anderson bound every later claim: across four hundred participants, interventions built to shape reliance generally fail to improve it [Bo et al. 2025]. Design intent alone does not warrant a cognitive outcome.

Added transparency does not reliably calibrate trust – trust tracking a system's true capabilities, in Lee and See's founding sense [Lee and See 2004]. Kizilcec finds a bell-shaped relation between transparency

and trust after an unexpected outcome [Kizilcec 2016]. Pareek et al. find any explanation attached to a credibility verdict raises reliance on it whether or not it is correct, because "users could not discern whether the AI directed them towards the truth" [Pareek et al. 2024]. A CHI'26 controlled study on a claim-evidence interface for scholarly Q&A sharpens the same hazard: displaying provenance lowered trust without changing reliance, and cost usability besides [Martin-Boyle et al. 2026b]. The same team's companion paper instead builds a twenty-pattern, seven-category taxonomy of LLM error in scholarly QA from practicing scientists' own strategies – a human-expert alternative to a mechanical pass/fail check [Martin-Boyle et al. 2026a]. None of these findings license a calibration claim for a display designed against, not tested for, that decoupling; our architecture makes no reader-facing trust or reliance claim for this reason. This literature builds and tests interventions at the interaction — a forcing function, a friction principle, a provenance panel, each for one occasion. None asks what changes when verification is a standing property of an architecture, not an addition a user can decline.

2.2 AI-for-research systems

Three *Nature*-published systems anchor the state of the art for AI-assisted research. Google's Co-Scientist orchestrates six agents around a "scientist-in-the-loop" paradigm, screening self-generated hypotheses by tournament before a wet-lab terminal check [Gottweis et al. 2026]; FutureHouse's Robin closes that loop end to end, recovering a genuine drug candidate [Ghareeb et al. 2026]. Stanford's Virtual Lab pairs a principal-investigator agent with specialist and critic agents, architecturally closest of the three to our own moderator/sub-agent/ratification pattern, though its object is open-web science, not a closed corpus [Swanson et al. 2025]. Each concedes what remains undone for a claim's provenance: Co-Scientist names enhanced provenance-tracing agents as future work [Gottweis et al. 2026]; Robin's hallucination check was manual and one-time, not a standing gate [Ghareeb et al. 2026]. A fourth system without Nature's imprimatur runs a comparable check: Sakana AI's AI Scientist screens papers through simulated review [Lu et al. 2024], its successor producing the first fully AI-generated paper to pass workshop-tier peer review [Yamada et al. 2025].

SPIRE brings a cousin of that architecture to the humanities. Seven agents realize the seven canonical research operations Unsworth's own invited talk first named as Scholarly Primitives discover, annotate, and — unlike the division of labor we describe later — themselves bind citations and synthesize argument over classical Chinese and Latin corpora [Pan et al. 2026; Unsworth 2000]. Its verification is generation-time and self-directed, scoring an essay against a gold benchmark once rather than as a check a reader could re-run. Its own account of the human's remaining part – scholarly judgment, teaching, and peer evaluation – addresses the scholar producing an argument, not a reader checking one delivered [Pan et al. 2026]. Narrower DH-domain terminal checks exist too: an LLM-assisted named-entity and cross-lingual annotation-projection check over a TEI/XML critical edition reaches over 97% projection accuracy [Galka and Vogeler 2026], and a framework embeds close reading into a retrieval-augmented-generation pipeline to improve its own output, not the reverse [Zhou et al. 2025]. Both target what a model produces while processing a source, not what a reader does with a delivered claim, the distinction this paper turns on. A preliminary Evidence-RAG peer-review workspace for a digital-history journal is closer still, retrieving evidence only from a submitted manuscript's own text and keeping the editor as final decision-maker, but its support judgment is a soft model label, not a mechanical verbatim check [Guérard et al. 2026]. Fang et al. embed simulated peer agents

into the reading interface, shifting critical-thinking behavior but leaving what the reader has offloaded unchecked [Fang et al. 2026]. Creative Reading raises the delegation risk from the reader's side, against tools that frame reading as "reading to discard"; its target is what augmentation should preserve, ours a mechanism for checking what is already offloaded, and the two complement rather than compete [Liu et al. 2026]. Berry names the nearest conceptual target without building it [Berry 2025]; Klein et al.: "Models make words, but people make meaning" [Klein et al. 2025].

A neighboring NLP literature runs a terminal check of a related kind, evaluated since ALCE established short-span citation attribution as this family's own benchmark standard [Gao et al. 2023b]. RARR edits unsupported spans out of a model's own output [Gao et al. 2023a]; FActScore decomposes a generation into atomic facts scored against a held source [Min et al. 2023]. Self-RAG critiques its own generation against what it retrieves [Asai et al. 2024]; SciFact verifies a scientific claim against a held evidence corpus [Wadden et al. 2020]; AVeriTeC extends the same task to real-world, open-web-checked claims, reporting substantial agreement on its own verdicts [Schlichtkrull et al. 2023]. Each scores a model's own generation against a corpus once, at generation time — not even a recent, human-directed, re-runnable citation-verification tool checking an open, unbounded citation graph [Haan 2025] — embeds that check in a governance process with a disclosed false-positive taxonomy over a closed, pre-manifested corpus. SciTrue is closer still: a peer-reviewed, human-evaluated claim-verification interface that nonetheless checks a submitted claim once against the open Semantic Scholar index, not as a standing property of a closed, pre-manifested corpus [Tan et al. 2026]. That family's checkers are now benchmarked: a preprint benchmark of thirteen citation-metadata verifiers finds that false-positive rate, not recall, decides whether a verifier is deployable [Reizinger and Brendel 2026] — independent support for our own disclosed false-positive defect taxonomy's premise. A cross-domain empirical result reinforces the same choice: across fourteen legal-citation model configurations, LLMs catch 93–100% of wrong-case corruptions but only 37–61% of wrong-pinpoint corruptions on court opinions — evidence that topical relevance and page-level support are conflated capabilities a mechanical, script-run check is designed to keep separate [Verma 2026]. Szeider's architecture bypasses the language model for the final export step entirely, resolving natural-language citation fragments against an authoritative bibliographic database directly [Szeider 2026], addressing citation-formatting accuracy rather than the quotation-verification step our own check targets. Rashkin et al.'s human-rated Attributable-to-Identified-Sources standard is this family's own foundational attribution test, its nearest peer-reviewed sibling to our binary verbatim-match check, though scoring overall support once rather than gating a specific offloaded step [Rashkin et al. 2023].

2.3 Provenance-tracing and claim-adjudication architectures

A distinct, more recent literature builds general-purpose provenance machinery, not a domain-specific check, surveyed broadly by Wang et al. [Wang et al. 2026]; none engaged in this paper before. F(AI)^{2}R extends W3C PROV-O with a seven-rung, human-gated verification ladder whose top two rungs no AI agent can grant [Krebs 2026]. LEDGER reconstructs a layered, typed-edge trace graph from an agent's coding session for a human to traverse, but verifies no claim, its own graph construction "not fully deterministic" [Kim et al. 2026]. Both cite the same closest prior work: a PROV-O extension binding an AI agent's own prompts, responses, and decisions into a scientific-computing workflow's provenance graph, typed throughout but scoped to an open, streaming corpus [Souza et al. 2025]. A similarly-named but unrelated system, LedgerMind, checks multimodal agent trajectories and implements a working invalidation mechanism propagating staleness

Table 1. Six components of a verification architecture, scored against the nearest systems in this literature. Y = present as the column asks; P = present but narrower or caveated; N = absent, including designed-but-not-shipped; – = not applicable (a vocabulary standard). Values are each system's own stated text, not inference. PROV-aligned scores whether a system's own relations are named after PROV-O/PROV-N terms, not whether the system has been checked against those standards; §6 reports a separate, disclosed generalization check, not a shipped pipeline component, that validated a full PROV-O/PROV-N serialization of our own corpus against five PROV-Constraints checks. Our own Invalidation cell scores P, not Y: implemented over the research pipeline's own ledgers (§6), not the deployed reading application, and with the full resweep retained as backstop. Our own PROV-aligned cell scores P, not Y: a live exporter emits PROV-O-named relations from the invalidation event log (182 triples, 36 events); five of five structural checks pass, following a same-round repair, generation-before-use remains not run.

System	Corpus closed	Exact-ness	Pred. typing	Invali-dation	Human gate	PROV-aligned
Our architecture	Y	Y	Y	P	Y	P
F(AI)^{2}R [Krebs 2026]	N	N	P	N	Y	Y
LEDGER [Kim et al. 2026]	N	N	N	N	Y	N
PROV-AGENT [Souza et al. 2025]	N	N	Y	N	N	Y
LedgerMind [Du et al. 2026]	N	P	P	Y	N	N
Evidence-Ledger Adj. [Chen et al. 2026]	N	N	P	N	P	N
ScientistOne/CoE [Meng et al. 2026]	N	P	Y	N	N	N
SPIRE [Pan et al. 2026]	P	N	N	N	P	N
Evidence-RAG JDH [Guérard et al. 2026]	P	N	P	N	Y	N
ProVe [Amaral et al. 2024]	N	P	P	P	Y	N
SemanticCite [Haan 2025]	N	N	N	P	N	N
SciTrue [Tan et al. 2026]	N	N	P	N	N	N
ALCE/RARR/FActScore group [Gao et al. 2023a,b; Min et al. 2023]	N	N	N	N	N	N
PROV/Web Annot./Micropub. [Clark et al. 2014; Lebo et al. 2013; Sanderson et al. 2017]	–	P	–	Y	N	Y

to dependent claims when a grounding entry is revised, a property our own tooling now also implements, differently scoped (§6) [Du et al. 2026]. Evidence-Ledger Adjudication classifies a claim against evidence into one of four relations at drafting time, benchmarked against 2,335 external rows, the check a single LLM judgment run once, not routed to a human [Chen et al. 2026]. ScientistOne's Chain-of-Evidence is closest on typed multiplicity, its four mechanized integrity checks reporting zero hallucinated references, but scoped to code and logs for one paper-writing session [Meng et al. 2026]. ProVe verifies a knowledge-graph triple against Wikidata text with a tunable threshold and curator-in-the-loop escalation, an older, non-agentic lineage for the mechanical, human-escalated pattern our own check descends from [Amaral et al. 2024]. Nearest of all is Jing's Traceable Scholarship, a normative framework for AI-assisted humanistic inquiry built on one Kant knowledge base. Its NO_EVIDENCE marker — an explicit refusal to answer past a scope boundary rather than fall back on the model's own knowledge — is the closest single mechanism in this literature to our own not-found disposition, independently arrived at [Jing 2026]. It discloses, though, no invalidation mechanism, no executed quantitative evaluation, and one case study authored and judged by its own builder. The W3C PROV family, the Web Annotation Data Model's quotation selector, the Micropublications claim-evidence ontology, and Nanopublications' three-part statement/provenance/publication-info structure supply this literature's vocabulary layer [Clark et al. 2014; Groth et al. 2010; Lebo et al. 2013; Moreau and Missier 2013b; Sanderson et al. 2017].

2.4 Positioning

Both bodies of literature above answer a different question than ours: what a system *makes possible*, evaluated by what a reader or curator can subsequently do with it, is a distinct and legitimate contribution type

from a measured behavioral effect [ACM SIGCHI 2026; Olsen 2007; Wobbrock and Kientz 2016]. One annotated-portfolio methodology names this outright: documenting a design's own rationale beside, not in place of, predictive theory [Gaver and Bowers 2012]. It is the type this paper claims. Table 1's six columns are not exhaustive, and not all trace to one source. Three follow this architecture's own commitments (§4) — closed corpus to D3, exactness to D1, human gate to D5 — while typed predicates, invalidation, and standards alignment are comparison-literature axes added here, not drawn from those five. Read across Table 1, every comparator matches our own architecture on at most a few of six columns, itself resting on the closed, pre-manifested corpus and typed, human-gated checks in §4–§5. Four of those differentiators carry a direct, re-derivable evidence locator, not only an architectural description. Corpus closure is a live, dated count (245 sources as of 2026-08-24, §4). Locator pinning is a re-runnable script, re-run at six rows and thirteen pins in this paper's provenance appendix, zero mismatches (§6). The human gate is reserved at named checkpoints with no timeout or default-approve path (§4–§6). Audit replay spans two instruments: a six-snapshot historical replay reproducing an already-published finding (§6), and a fourteen-pair retroactive replay of the invalidation engine against real committed history, exact on every pair including its strongest case, a nine-row identity re-point – corroborating detection, not independently the propagation logic (§6). PaperTrail is not a row here: it measures a display's effect on a reader's trust and reliance, not a verification architecture's components, and its finding is addressed in Discussion [Martin-Boyle et al. 2026b]. F(AI)^{2}R is stronger on the human-ratification ceiling and on standards alignment, both properties our account does not yet claim; ScientistOne's Chain-of-Evidence is stronger on typed multiplicity; SciTrue is stronger on peer-reviewed, human-evaluated attribution accuracy, a property outside this table's six columns. Jing's Traceable Scholarship is nearest on domain; Co-Scientist's and Robin's terminal check is stronger for their object than ours; SPIRE performs a synthesis we withhold by design. A closed corpus, a re-runnable mechanical check, and a reserved human gate, applied together to AI-assisted research production in the humanities, is the combination this literature has not yet put together; our contribution is that combination, applied to the offloaded step it does not yet check as one. The Evidence-RAG JDH workspace comes closest, matching the human-gate leg and approaching the closed-corpus one, but for a different task, editorial peer review, and with a soft model judgment where the third leg calls for a mechanical one [Guérard et al. 2026]. The comparison above draws mostly from NLP fact-verification and knowledge-graph provenance work, not CHI systems literature.

3 Scope

This paper reports a Type-3 contribution [ACM SIGCHI 2026; Wobbrock and Kientz 2016]: an architecture and its own bounded self-application, not a controlled evaluation of reader outcomes. Every check reported below verifies a mechanical or structural property – that a string occurs in a held source, that a citation resolves, that a claim inherits a linked quotation's status – never a semantic entailment, an independent cross-model judgment, or a generalization beyond what was sampled. Each such boundary is stated at the point in the argument it bears on, not collected separately from it.

4 The Verification-Gated Architecture

The gaps named above translate into five desiderata a verification-gated architecture must satisfy, stated as design commitments; none is a claim about what any reading occasion demonstrates.

D1. Verification must be mechanical, not confidence-based. Self-reported reductions in critical effort under generative-AI use track confidence in a system's output, not its correctness [Lee et al. 2025]; trust calibration is itself a metacognitive demand a design satisfies structurally or leaves for the user [Tankelevitch et al. 2024]. The desideratum: a check run against a record outside the model's own assertion, never a score it reports about itself.

D2. Offloading must be checkable, not silent. A model unable to signal its own uncertainty is a documented hallucination failure mode [Ji et al. 2023], and interface guidance for human-AI systems calls for making clear what a system can and cannot do [Amershi et al. 2019]. The desideratum: every offloaded step carries a visible pass or fail against an outside record, with a not-found result treated as a first-class outcome. This gate runs the same check regardless of reader expertise, a choice made in view of, not blind to, evidence that identical assistance helps experienced readers and hurts novices [Bae et al. 2026]; adapting it to expertise is future work the reliance literature counsels caution about [Bo et al. 2025].

D3. The corpus must be closed, manifested, and audited, not the open web. Multi-agent systems built for autonomous synthesis over an open literature [Pan et al. 2026] answer a different question: not what can be synthesized from everything available, but whether a fixed, itemized, deduplicated source set supports a specific claim. A check is only as strong as what it checks against, a design assumption rather than an established reliability advantage. "Closed" means closed at each audited snapshot, not fixed for the project's life: the held corpus has grown at every intake cycle (245 sources as of 2026-08-24), under the same gate as everything checked against it. Citation practice outside STEM publishing often lacks a comparable metadata norm and can dispute which edition, translation, or witness a quotation belongs to, a dispute this check leaves unadjudicated. It makes auditable only which held witness matched, not which one is correct.

D4. Challenge and contest must be designed in, not hoped for. Users of a multi-agent tool have asked, unprompted, whether it will ever push back on them [Adavi et al. 2026]; surfaced provocations have been shown to restore critical engagement unprompted AI assistance otherwise erodes [Drosos et al. 2025]. The desideratum: a reviewing role independent of the work it reviews, with a defined severity scale and a verdict that can withhold sign-off.

D5. Interpretive judgment must be structurally reserved for the human. Human-machine task allocation by comparative advantage – mechanical work to the machine, judgment to the person – is the original framing this architecture extends [Engelbart 1962; Licklider 1960]. This is narrower than a claim that only two kinds of decision exist; most work in §5 blends mechanical and interpretive elements. For one class specifically – what an argument claims, when a source may be trusted, when a draft is ready – no mechanical check and no agent verdict, at any model tier, may resolve the question in the reader's place.

Cognitive load theory holds working memory's capacity to relate information at once sharply limited [Kirschner et al. 2006]: low-interactivity material is tractable where the same material assimilated as one whole is not [Sweller 1994]. D1 and D4 apply this twice, an inference, not a measurement.

Offloading needs a construct, not only a name: an epistemic action is "a physical action whose primary function is to improve cognition by" cutting the memory a reasoner must hold and the probability of error

Table 2. A narrow predicate registry: ten of the checks the verification-gated architecture runs, typed by what decides each one. No row claims semantic or entailment verification – the registry contains none.

Check	Input	Decided by	Type	Passing does not prove
Quotation match	a quoted span	normalized character-sequence containment (case-, punctuation-, and whitespace-insensitive)[1]; a 0.95–0.99 near-match is disposed `FUZZY`, reported separately, never counted as `VERIFIED`	exact	completeness; entailment of the surrounding paraphrase
Witness resolution	a citation key	a five-rung fallback ladder	structural	the resolved source is right beyond string agreement
Extraction selection	a resolved source	a ten-tier variant preference order	structural	the chosen variant is itself defect-free
Claim inheritance	a claim's evidence link	a plain-string join to quotation status	structural	the claim's own prose paraphrases faithfully
Citability tier	source metadata	a four-tier publication rule	structural, agent-run	the tier assignment itself is error-free
Manifest dedupe	a hash, DOI, title-year	three-way exact match	structural	two entries are not the same work under a different edition
Locator placement	a (claim, location) pair	structural plus clause-boundary check	structural	the prose at that location says what the row claims
Fault-injection catch rate	sampled verified rows	four corruption families re-run through the match check	structural, sampled	robustness against corruption types never sampled
Gate and AI-review finding	a full draft	severity-graded reading, no fixed procedure	human-gated	correctness beyond that reviewer's own judgment at that reading
Director ratification	a proposed decision	a dated, human-authored record	human-gated	the ratified decision is correct beyond who decided

[Kirsh and Maglio 1994]. Clark asks "in what ways designers and users of new tools and technologies might exercise due epistemic caution" [Clark 2015]. D1 and D2 answer that at one step; D3 through D5 chain the steps so no later stage rests on one never checked.

A narrow predicate registry. These commitments are enforced by named, individually scoped checks, not one undifferentiated "verification." Each is typed as one of three kinds, the distinction §6 applies throughout. *Exact* means a script alone resolves it; *structural* means a script resolves it by a fixed rule over a relation a human designed; and *human-gated* means no script resolves it, because what is decided is not the kind of thing a rule decides. Table 2 lists ten such checks – a designed mechanism individuates more easily than a found one [Smart 2022, 2024]. Its last column weighs as much as its operator column: what a pass does *not* establish is part of the check's specification, following safety-engineering practice, where a sufficiency argument accumulates typed evidence rather than proving a claim [Hawkins and Kelly 2012].

Each check returns one of a small fixed label set, never a continuous score (`VERIFIED`/`TEXT-VERIFIED` for a match, `FUZZY`/`NOT_FOUND` for a failure, `NOT-VERIFIABLE`, `NO-KEY`, `MISMATCH`, `AUTHOR-RULED`). Three decision points are reserved for the director alone, closeable by no script or agent at any tier (§5).

[1]Implemented as containment on an NFKD-normalized, alphanumeric-only string (`predicates.py` ll.62–74, 105–121); a hit is labeled `VERIFIED` by convention, not measured similarity.

Borrowed vocabulary, not a borrowed standard. The registry's two most load-bearing relations already have names in an existing standard: a quotation's link to its source is PROV-O's `wasQuotedFrom`; a claim's reliance on a linked quotation is `hadPrimarySource` [Lebo et al. 2013]; a locator pin's pointer to prose lines is the Web Annotation Data Model's `TextQuoteSelector` [Sanderson et al. 2017]. Neither standard checks the relation it names: PROV-O asserts without verifying verbatim; `TextQuoteSelector` locates without verifying faithfulness. The match and placement checks supply the confirmation each presupposes.

Invalidation, implemented as pipeline tooling. A persisted, per-quotation content-hash engine tracks which claim rows depend on which extraction-variant file, the way a build system's task graph would [Mokhov et al. 2018], marking a dependent row `STALE` until a predicate re-run or a dated human ratification clears it, with a disclosed circularity limit. The cycle-boundary full resweep remains the unconditional backstop, and the deployed reading application carries no comparable `stale` state.

A static, line-level audit tested this correspondence against all twenty architecture elements – five design commitments, ten predicates, and five further items including the seven-gate structure – rather than asserting it. One mechanism was confirmed (a database constraint on human-only editorial actions, cited twice); ten were partial analogs; eight, among them the seven-gate structure and D3's closed-corpus commitment (whose discovery layer exposes open-web lanes), had no app-side counterpart. This establishes static correspondence only, not that matching code paths produce matching verdicts on real input. The reference implementation is a specific case of this same method – checkable, dependency-ordered offloading across narrowly scoped, reviewed agents – run by the build agent for a reader instead of by this method for the director. One commitment governs both, rather than two designs sharing a vocabulary by coincidence. Figure 6 schematizes that implementation's own verification-relevant panels; the reading experience they compose is not itself what §7 evaluates.

5 Pipeline Architecture

A single human director oversees two parallel top-level agents, each authorized to delegate task-specific work to further sub-agents. The paper-writing agent conducts the research program and drafts the paper the reader now holds; the parallel build agent constructs the reference implementation §7 draws on solely to demonstrate how the method aids research. That implementation is this method's own build-line output, not a second system under separate review; Figure 6 schematizes only the panels carrying its verification apparatus, and Figure 1 situates this division inside the gate structure below. The boundary between the two lines is read-only and was checked, not merely declared: a version-control snapshot of the implementation repository, taken before research work began, already shows that repository in a modified, actively developed state – dated evidence the build side was already at work on its own schedule. The two lines feed each other only through the director's own decisions and discrete authored handoff artifacts, whose authored-and-ready status is what the record documents – whether the build agent applies a comparably disciplined process on receiving one is the build side's own account, not independently checked here. The two lines are not independent in a statistical or epistemic sense, sharing one director, project, corpus.

Within each line, delegation is tiered by task complexity: lightweight models handle metadata extraction and formatting, mid-tier models handle annotation and citation mining, top-tier models handle synthesis, adjudication, and adversarial review. Every delegation is logged by role, tier, and task specification, and every

Table 3. The seven-gate structure. No gate is self-certified: each pass condition is either a mechanical check or a review role distinct from the work's own author.

Gate	Checkpoint type	Pass condition	Reviewing party
0	Scaffold	Directory structure, status tracker, and version control initialized	Moderator
A	Reference audit	Reference material fully inspected with no write access exercised	Moderator
B	Source intake	Every candidate source metadata-verified, tiered for citability, and deduplicated before manifest entry	Intake agent; moderator-reviewed
C	Literature review	Each disciplinary review verified against its search brief	Discipline-review agents
D	Thesis selection	Central argument selected and ratified by the human director; independent sign-off granted	Human director; independent reviewing agent
E	Mechanical sweeps	Instruction-leak, provenance-leak, tense, effect-verb, quotation, and typography checks all report pass	Automated sweep tooling
F	Adversarial review	Severity-graded review returns a verdict of pass, or sign-off is explicitly granted	Adversarial-review agent; human director

sub-agent's output passes to a distinct reviewing agent before any later stage builds on it — answering the desideratum that offloading be checkable, not silent (D2, §4) [Amershi et al. 2019; Ji et al. 2023]. The tiering converts a per-task metacognitive burden Tankelevitch et al. [2024] identify into a one-time architectural decision, warranted by the comparative-advantage framing under D5 [Engelbart 1962; Licklider 1960]: even the top-tier synthesis agents remain sub-agents under human-ratified review, not autonomous authors.

Work does not move forward on a sub-agent's own say-so. It clears a sequence of seven named gates, Gate 0 through Gate F, specified in Table 3. No gate is self-certified: a gate's status (open, passed, passed with conditions, reopened) is a recorded fact about the project's history, not assumed from the fact that work continued.

The pipeline is cyclical, not sequential: a cycle closes on a full adversarial review and, where that review or a later ruling calls for it, reopens (Figure 1). Figure 3 diagrams this cycle in full. Two episodes evidence this operating as a check, not a formality, full detail in Appendix B. The full sequence, including both loop-backs and the sub-process detail behind Gates B and E, is traced as a flowchart in Figure 2. The sibling manuscript – a companion, humanities-focused paper this project produced on a paused, separate publication track – saw a reopened Gate D and a Gate F review recorded as convened but not passed across two cycles. This paper's own review trail runs the same gates, traced round by round in the appendix (Figure 8, Appendix B), kept current each round. The mechanical checks answer to the same discipline: one check in §6 was found defective, repaired, and re-run four times – a check never re-checked can accumulate exactly the drift the gate structure exists to prevent. Closure is reliable at the round-pair level – a self-report paired with independent re-verification, not the self-report alone. Self-reported closure regularly overstates its own completeness, catching as few as ten of twelve independently re-checked claims in one governance report – a pattern caught only by the paired pass.

The governance record requires a standing AI-review stage between the adversarial review and the director's own read, run either in-project or by an author-commissioned reviewer independent of the drafting

agents and tooling – a different model family, under an absolute no-edit rule. The stage's only output is a severity-graded findings report, never a revision, so it sits outside the seven-gate count: even a clean report grants no ratification. Two external-vehicle audits have run to date: the first on this paper's third draft, intaken as a provenance-documented package under a governing prompt; the second, same model family but broader project access, on Cycle 8's v21b build, thirty-one findings dispositioned this round. Its findings, set against the pipeline's own seven-dimension review of a later, fourth draft, overlapped at a Szymkiewicz–Simpson coefficient reported in full in Table 11 (Appendix B), a comparison spanning a version gap that biases the measured overlap downward. Every blocking finding was raised by both reviews, though the internal panel independently re-derived only about a third of the external register's findings overall – both readings accurate, the misses concentrated in the minor tail. Both aside, every check above is designed, run, and mostly reviewed by agents inside the project it audits.

Where AI co-scientist systems select among machine-generated hypotheses through an Elo-scored self-play tournament [Gottweis et al. 2026] or an LLM-judged pairwise ranking [Ghareeb et al. 2026], this project applies the same competitive-selection logic to a document. Three prompted personas, run against one model family rather than three independently sourced models, scored three candidates for a piece of scope; the surviving candidate was repaired against its sharpest critic's flaws and grafted with material its rivals contributed. Model-family diversity alone does not restore independent errors once judges share a prompt and a corpus [Kohli 2026]; this panel's three scores are read as correlated readings from one model, weaker than the AI-review stage above holds itself to. A self-critique ladder is excluded by the same non-self-certification rule that gives Gate F its bite; hypothesis evolution is excluded because this method's rule is the structural inverse – the model indexes and verifies but never originates a claim (§6).

Before any source may support a claim, it passes a source-intake pipeline (D3, §4) running three checks in sequence. The first verifies metadata against the document itself, never a filename or citing paraphrase; the second classifies it into a four-tier citability rule (peer-reviewed work, primary sources, work accepted for publication, preprints), everything outside those tiers held background-only. The third deduplicates at the work level, by hash, identifier, and normalized title-and-year, before registration in parallel manifests that must never diverge. The resulting corpus is closed, manifested, and audited rather than assembled ad hoc from the open web, the condition the next section's verification claims depend on.

Every substantive ruling that shapes the argument — a directive from the director, a moderator's resolution of an open question, a change of thesis or scope — is entered in a dated, append-only record naming who decided, the rationale, and the row's current status. As of 2026-08-24, the record holds two hundred fifty-eight dated entries, one hundred nine opening by naming the director, and a separate task ledger of three hundred ninety-one rows (Appendix B). A superseded ruling is marked superseded, not deleted, so a reviewer can verify a reopened gate really was reopened on a logged ruling, not redrawn after the fact (D4).

None of this closes the loop mechanically. A small number of checkpoints are held out, by design, as points a mechanical pass or an agent's own synthesis cannot clear alone: adopting a thesis, changing a framing that alters the argument's central claim, and signing off on a finished draft as ready to stand behind. Each requires the director's own dated confirmation, distinct from an agent's verdict at any model tier — the structural answer to what Passi and Vorvoreanu describe as oversight functioning as false security [Passi

and Vorvoreanu 2022], routed through the director by construction so the pipeline's terminal state cannot self-certify (D5). One request's path through all seven gates is traced in Figure 4.

6 Evaluation and Results

Three operations share the word "check" below. A quotation check is a mechanical, script-run verbatim-substring match against held text. A claim check is a rule-based, non-mechanical test of whether a claim inherits a linked quotation's status. A gate review is a severity-graded, human-and-agent judgment call, not itself decidable in the sense the first two are (§5 details the gate sequence).

6.1 Method: A Frozen Evaluation Snapshot

The results below run against a fixed *frozen evaluation snapshot* (Figure 5): a pre-registered, hash-verified evaluation protocol, frozen 2026-08-22 (RNG seed `20260822`), before any measurement ran. That protocol fixes the two mechanical ledgers and the source manifest at 748 quotation-ledger rows, 693 claim-ledger rows, and 193 manifest entries. The corpus's live size differs – as of 2026-08-24, 245 manifested sources, 792 quotation-ledger rows, 730 claim-ledger rows – ordinary intake growth, not a ledger repair. A companion 37-case fixture suite tests the checking machinery itself: 33 passed, zero failed, four informational disclosures – a data-quality census, not a pass/fail result.

6.2 Results: Seeded Fault Injection

A pre-registered mutation catalogue, in the mutation-testing tradition [Papadakis et al. 2019], corrupted a fresh, randomly seeded sample of the frozen ledger's verified rows and re-ran all fifteen classes (223 mutants total) through the quotation check; Table 4 reports the five classes carrying the interpretive stakes. It reports Wilson 95% intervals [Brown et al. 2001; Wilson 1927] rather than a bare fraction, since boundary cases here (thirty of thirty, one of thirty) are exactly where a normal-approximation interval misleads. Two cheaper baselines – trust a citation because it resolves to a manifested source; trust the ledger's recorded status without re-inspecting the quoted string – ran against the identical sample and caught nothing at every class, confirming that detection requires inspecting the quoted text.

Table 4. Seeded fault injection against the frozen ledger, capability and negative-control classes ($n = 30$ drawn per class, seed 20260822). Wilson 95% confidence intervals; both baselines (citation-exists-only; trust-the-stored-status) caught 0.0% at every class, Wilson upper bound under 12%, confirming neither substitutes for re-inspecting the quoted string.

Class	n caught/valid	Rate	Wilson 95% CI
Fabricated quotation	30/30	100.0%	[88.6, 100]%
Polarity reversal	29/29	100.0%	[88.3, 100]%
De-hyphenation toggle[a]	29/30	96.7%	[83.3, 99.4]%
Curly-quote toggle	29/30	96.7%	[83.3, 99.4]%
Truncation (negative control)	**1/30**	**3.3%**	**[0.6, 16.7]%**

A truncated quotation is, by construction, still a genuine substring of the source. A check answering whether quoted words occur in order cannot distinguish a complete quotation from a shortened one, so the one-in-thirty catch here is noise around a structural zero, not a near-miss. It is reported at the same prominence as the higher rates, not aggregated away. That the checker reliably catches transpositions,

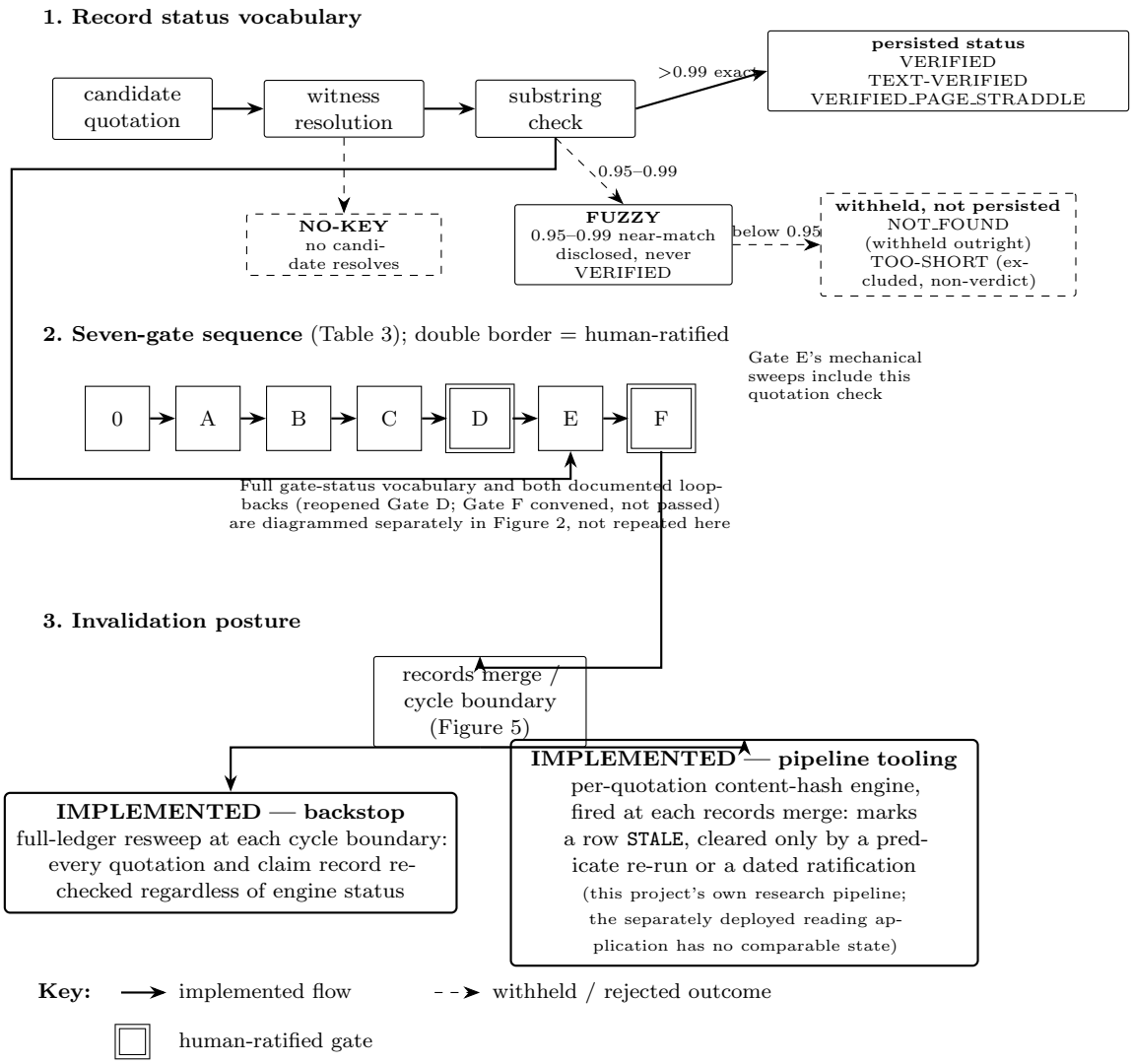

Fig. 1. Contract state and invalidation flow. Band 1 shows the record status vocabulary a checked quotation can reach: a candidate quotation passes through witness resolution and a substring check. An exact match above 0.99 similarity persists as one of three accepted statuses: VERIFIED, TEXT-VERIFIED, or the documented page-straddle limitation VERIFIED_PAGE_STRADDLE. A 0.95–0.99 near-match instead persists as the weaker, separately-reported FUZZY disposition, disclosed and never counted as VERIFIED. Anything below 0.95 is withheld outright as NOT_FOUND or excluded as a non-verdict (TOO-SHORT), and an unresolved witness instead returns NO-KEY. Band 2 places this check inside the seven-gate sequence of Table 3, with the two human-ratification points, gate D (thesis selection) and gate F (terminal adversarial review), drawn with a double border. The full gate-status vocabulary and both documented loop-backs are diagrammed separately in Figure 2 rather than repeated here. Band 3 shows the two invalidation postures this project's own research pipeline runs. The full-ledger resweep, unconditional backstop at every cycle boundary, re-checks every record regardless of engine status. A second, additive layer, a per-quotation content-hash engine fired at each records merge, marks a row `STALE` when an upstream input changes, cleared only by a predicate re-run or a dated human ratification; both postures are drawn IMPLEMENTED, distinguished in words rather than by line style. Neither posture describes the separately deployed reading application, which carries no comparable `stale` state (Figure 5 depicts the research-only fixture suite that first gave this engine's design a concrete artifact, predating this promotion).

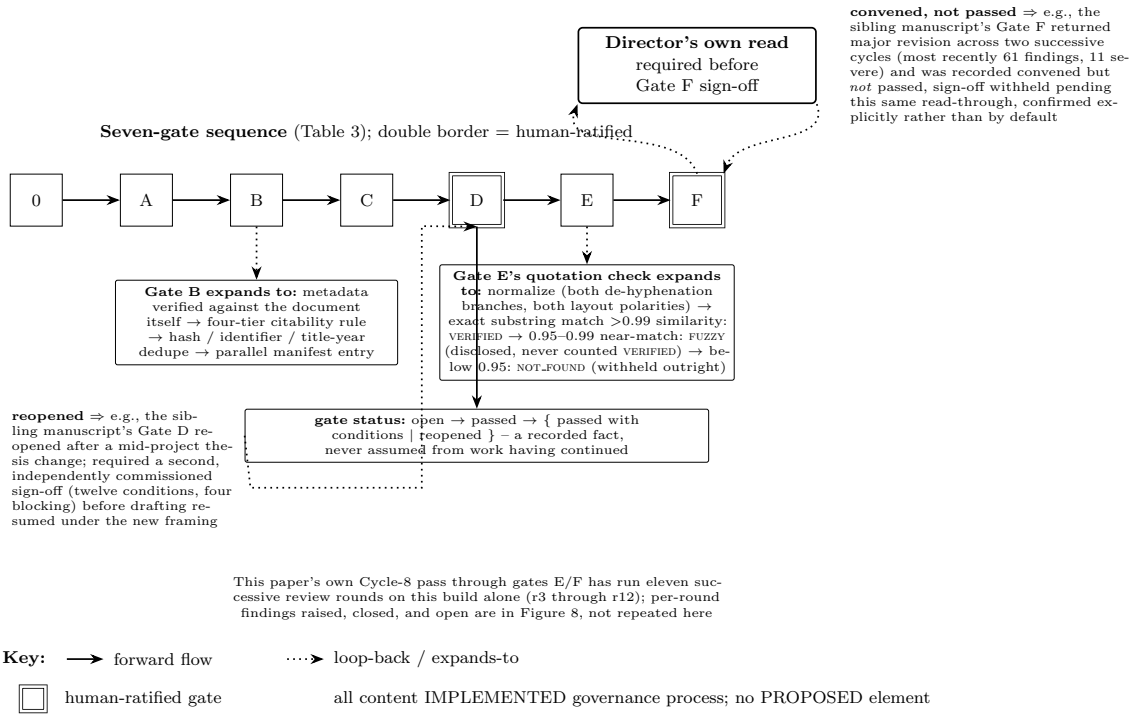

Fig. 2. The seven-gate pipeline as a flowchart, showing the forward path, both documented loop-backs, and the two sub-processes elsewhere described only in prose. Gate B (source intake) expands into its own four-step check (metadata, citability tier, dedupe, manifest entry), and Gate E's quotation check expands into its own normalize-then-match ladder. That ladder ends in one of three outcomes: an exact match above 0.99 similarity persists as VERIFIED; a 0.95–0.99 near-match persists as the weaker, separately-reported FUZZY disposition, never counted as VERIFIED; anything below 0.95 is withheld outright as NOT_FOUND (Table 2's Quotation-match row). Two documented loop-backs are drawn from this project's own governance record of its sibling manuscript, the only two cases on record where the pipeline reopened rather than proceeded. A reopened Gate D followed a mid-project thesis change that required a second, independently commissioned sign-off before drafting could resume. A Gate F was recorded as convened but not passed, sign-off withheld pending the director's own read-through, a status distinct from and not assumed from the fact that work continued (Appendix B, Sibling-Manuscript Gate Episodes). This paper's own Cycle-8 pass through gates E and F ran eleven further review rounds on a single build, detailed separately in Figure 8 rather than repeated here.

polarity reversals, and normalization edge cases says nothing about whether it catches a more complex or compound corruption never sampled. The coupling-effect assumption mutation testing relies on is itself untested here, an open premise, not a proven one; a same-round compound-corruption test found a 30/30 catch on two composed pairs, narrowing but not eliminating this premise.[2] A subsequent extension, run under a versioned protocol addendum (v1.2, frozen 2026-08-23), reran this instrument at $n = 100$/class, an 80-item equivalent-mutant audit, and 150 metamorphic-invariance checks across four transforms, with no class's threshold disposition changing and zero metamorphic violations (subsection A.3). As with Table 5's reference tool, protocol and implementation share the same author; an independent reviewer's blind re-classification

[2]The de-hyphenation-toggle class as generated tests a quote-side proxy for the branch property; a dedicated predicate-suite fixture above tests the property itself.

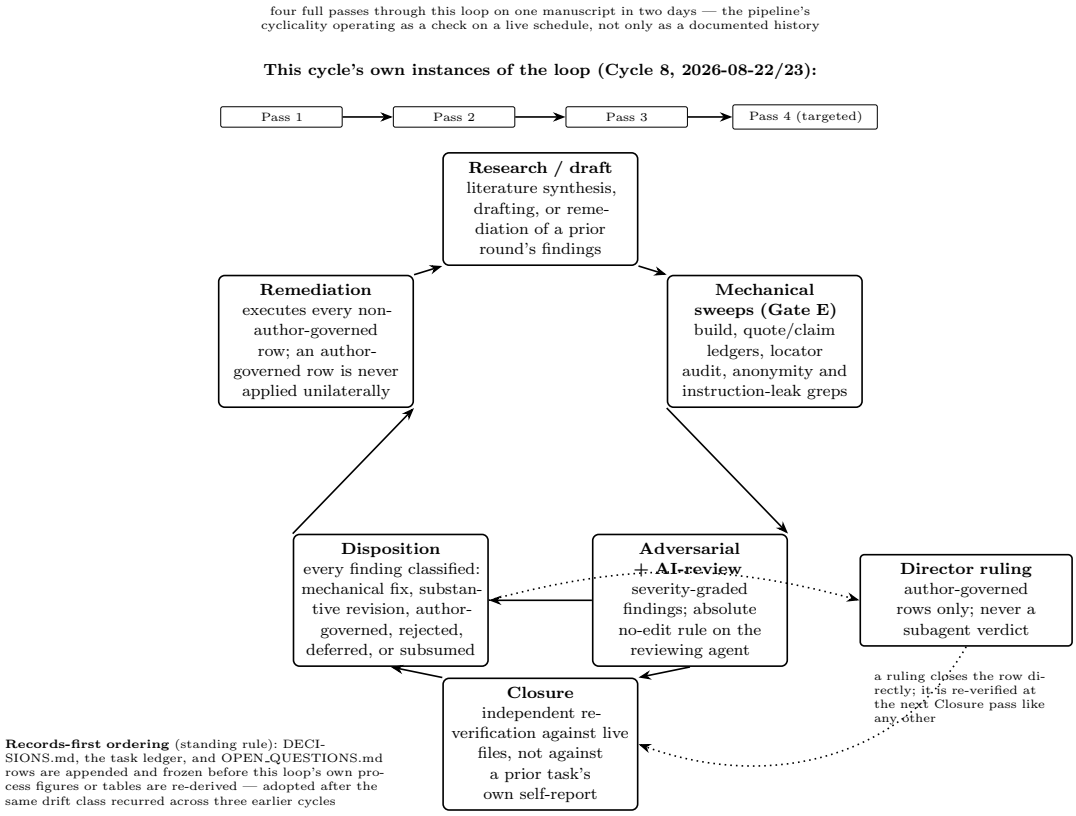

Fig. 3. The cyclical method, merging the project's cross-cycle research loop with the within-cycle review→disposition→remediation→closure loop the audit found to be the same mechanism at two granularities rather than two distinct processes. Six stages run clockwise: research or draft, mechanical sweeps at Gate E, an adversarial-plus-AI-review pass, disposition of every finding into one of six classes, remediation of every non-author-governed row, and an independent closure pass that re-verifies against live files rather than trusting a prior task's self-report. A findings row classified author-governed routes instead to a director ruling, off the main loop, and only a ruling — never a subagent's own verdict — can close it; a ruling is itself re-checked at the next closure pass like any other row. The top band names this cycle's own first four passes through the loop, all against one manuscript inside two days, each dated in that round's own review files; the round-by-round identifiers themselves, and the further rounds that ran against the same manuscript before submission, are relocated to Figure 8. The bottom band states the records-first ordering rule this loop runs under: governance rows are appended and frozen before this loop's own figures or tables are re-derived, a rule adopted after the same drift recurred across three earlier project cycles.

of 20 of the 80 audited items found 19/20 (95%) agreement, the one disagreement – a double-negative item the reviewer calls debatable but defensible – changing no reported rate.

A second, hand-built set of classes – a fabricated attribution, a source-version mismatch, a citation left unchanged while its claim is strengthened or reversed, a source reclassified after a claim rests on it – all pass silently. None of the three checks tests whether a claim's prose is entailed by what it cites. Two structural classes behave as designed: deleting a claim's evidence link reclassifies it from linked to unlinked with no false retention (thirty of thirty); removing a manifest source resolves every dependent citation to unresolved. A third surfaces a population artifact, not a checker regression: only ten of thirty randomly drawn manifest

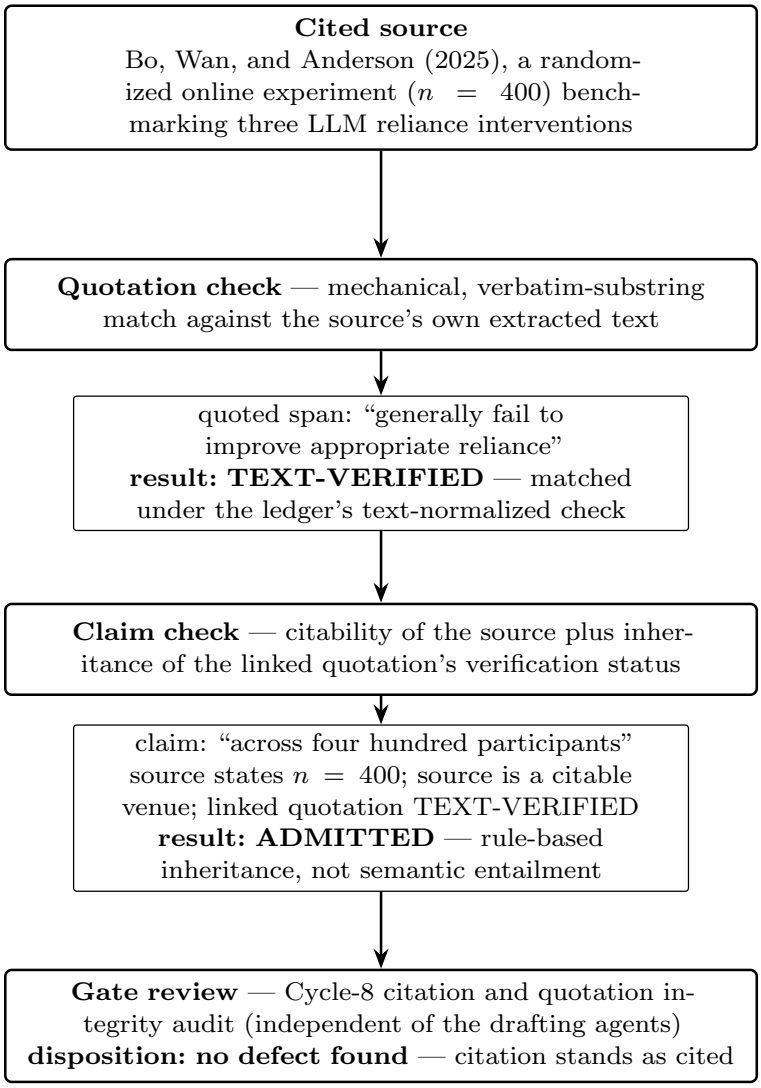

Fig. 4. One worked quotation-to-claim-to-gate path, illustrated with an already-cited, TEXT-VERIFIED example from the Cycle-8 citation and quotation integrity audit rather than a constructed illustration. A quotation from Bo, Wan, and Anderson (2025) is checked by a mechanical, verbatim-substring match against the source's own extracted text and returns TEXT-VERIFIED, the quotation ledger's status for a match confirmed under its text-normalized extraction path rather than the stricter default VERIFIED tier. A claim resting on that source's reported sample size inherits the quotation's TEXT-VERIFIED status and the source's citable classification, and receives an ADMITTED verdict under the claim check's rule-based inheritance test, which treats either match tier as sufficient and never asserts semantic entailment. The pair then passes through the gate-review layer, here the Cycle-8 audit that independently re-checked every quoted span and its attendant claims against the primary source and found no citation-integrity defect in this pair. Each of the three predicates — the mechanical substring match, the rule-based inheritance test, and the severity-graded gate review — is explicitly typed apart from the other two, per this architecture's central design commitment, rather than folded onto one ordinal ladder.

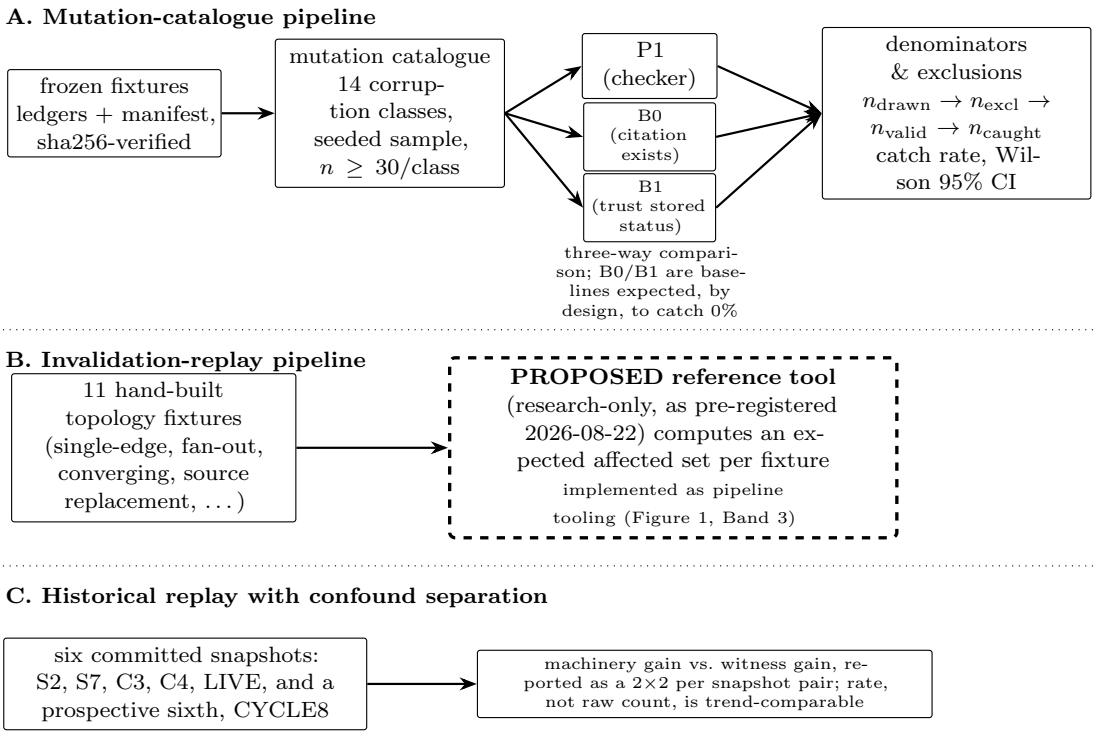

Fig. 5. The evaluation design this project's Cycle-8 protocol specifies, pre-registered before any measurement ran; results against this same frozen design are reported in Table 4 (Row A), Table 5 (Row B), and Table 6 (Row C), so this figure depicts the design those results were measured against rather than duplicating them. Row A: a seeded, sha256-verified frozen snapshot of the ledgers and manifest feeds a fourteen-class mutation catalogue (fabricated quotations, truncations, polarity reversals, and structural gaps such as attribution error and stale status among them), sampled at a minimum of thirty draws per class. The corrupted sample is scored under a three-way comparison — the real checker (P1) against two baselines, citation-exists-only (B0) and trust-the-stored-status (B1), both of which are expected by design to catch nothing. Every catch rate is reported with its denominators, its logged exclusions (normalization-equivalent, too-short, or malformed constructions), and a Wilson 95% confidence interval rather than a bare fraction. Row B: eleven hand-built graph topologies, each representing a distinct dependency shape a source, quotation, or claim edit could propagate through, are scored against a PROPOSED, research-only reference tool, pre-registered 2026-08-22, that computes the expected affected set per topology. This tool is the historical basis for the pipeline tooling implemented and detailed in Figure 1 (Band 3) and §6; at the time this frozen design was registered, neither this tool nor the live application implemented any invalidation propagation. Row C: the protocol extends the project's five-snapshot historical replay with a sixth point and requires the run to separate machinery gain from witness gain as a full two-by-two per snapshot pair. It also requires the run to compare rates rather than raw ledger-row counts, since the ledger itself grows across the series independent of any quality change.

entries carried a live citation key at all, nine of which resolved. That low raw rate reflects how much of the manifest a single paper cites, not a failure among entries actually cited.

6.3 Results: Dependency Invalidation, Implemented as Pipeline Tooling

The pipeline's re-verification is a full-ledger resweep at every cycle boundary and, between boundaries, a selective, dependency-triggered re-check of only the records a change touches – for this project's research

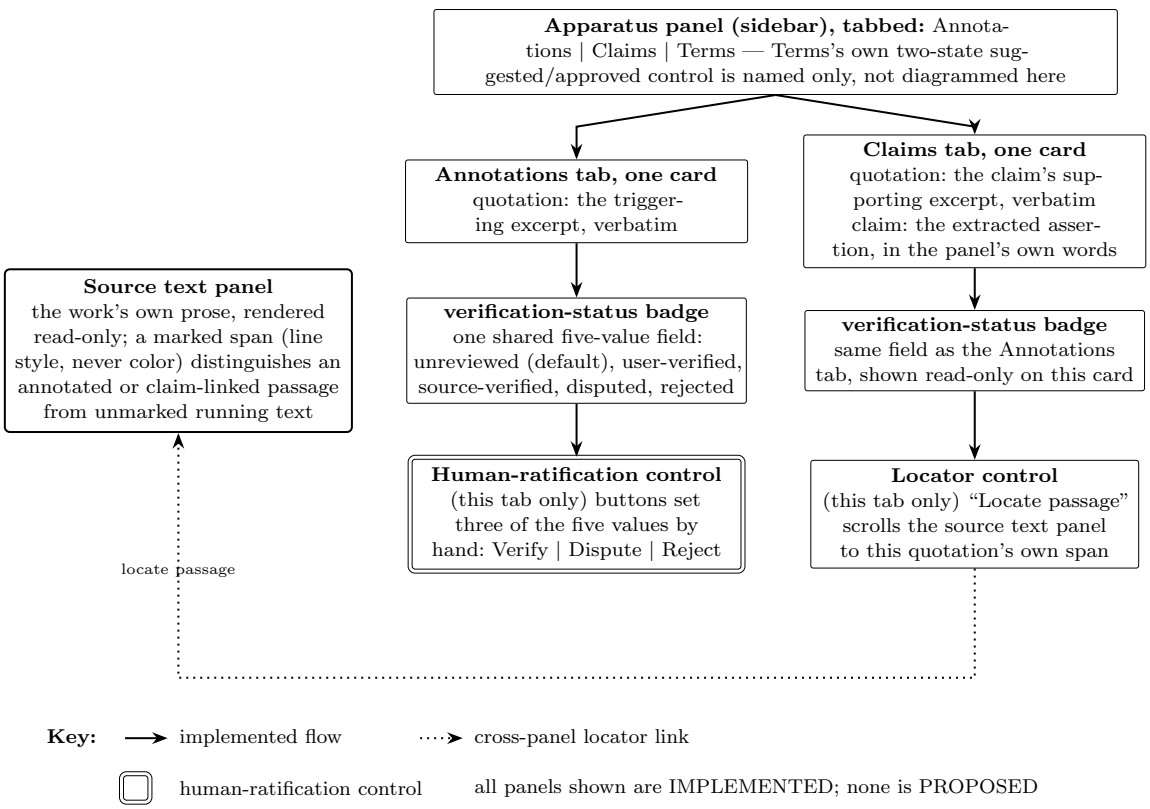

Fig. 6. The reference implementation's verification-relevant panels, as implemented, derived by direct inspection of its own front-end component code rather than drawn from a screenshot – no color, layout proportion, or literal screen composition is reproduced, only panel structure, control affordances, and status vocabulary. The source text panel at left renders the work's own prose read-only; an in-line marker, never color alone, distinguishes an annotated or claim-linked span. The apparatus panel (sidebar) carries additional non-verification tabs not shown here – Critical notes, a separately-named Apparatus tab, Sources, and My notes. Of its verification-relevant tabs, two are shown side by side, disclosed here as two separate cards rather than one combined view, since that is how the live interface actually separates them. The Annotations tab's card carries a verbatim triggering excerpt and, below it, this architecture's five-value verification-status field — unreviewed by default, then user-verified, source-verified, disputed, or rejected. Only three of the five (Verify, Dispute, Reject) are settable by a button on this same card, a distinction the figure states explicitly rather than implying all five are equally reachable by hand. The Claims tab's card instead pairs a verbatim supporting excerpt with the extracted claim text and the same shared status field shown read-only. It also carries a locator control ("Locate passage") that scrolls the source text panel to that quotation's own span, drawn as a dotted cross-panel link back to the leftmost panel. A third tab, Terms, exists in the same apparatus panel with its own, differently-scoped two-state suggested/approved control and is named but not separately diagrammed. Every element shown is IMPLEMENTED in the reference implementation as inspected; this figure depicts no PROPOSED construct.

pipeline, not the deployed reading application. A live engine computes a persisted, per-quotation content hash at every records-merge event and marks a row `STALE` when an upstream input changes, cleared only by a predicate re-run or a dated human ratification. This is the `wasInvalidatedBy`-style mechanism the provenance standard's definition calls for [Moreau and Missier 2013a]. The cycle-boundary full resweep is retained unconditionally as backstop, re-checking every row regardless of engine status; the engine narrows only the interval between resweeps.

The synthetic twelve-fixture suite that first gave this gap a concrete artifact (Appendix B) is unchanged: twelve of twelve exact-match – same-author, a from-specification confirmation, not independent validation. A different check draws on real history: a retroactive replay against fourteen real, git-verified commit-pairs, reconstructed from `git log`/`git show`, exact-matches every pair – 1,886 of 1,886 affected rows flagged, zero missed, zero spurious, precision and recall both 1.0 (Table 5). That replay's ground truth is computed by the same code the engine under test runs, corroborating row-level detection against real `git diff`s, though not independently the downstream-propagation rule. The strongest case, a nine-row citation re-slugging already ruled on, scores TP=9, FN=0, FP=0. An independent review, run pre-approval, found and fixed one conformance bug in its propagation logic, re-verified against both suites and the strongest real case unchanged. A live records-merge smoke test against the project's last real commit flags 110 triggers, 1,412 unexplained rows, and 18 correct `STALE` transitions – the round-10 `citable_status` repair propagating to 10 dependent claim rows.

This replay's population is bounded to git-committed history through this project's last real commit, excluding Cycle 8's uncommitted ledger growth; the removal/orphaning path has zero real historical exemplars: the one removal event's dependents were deleted in the same commit, resting only on the fixture suite and two unit-test regressions. The protocol's hash-mismatch `ERROR` rule is implemented, tested (thirty-one of thirty-one unit cases), and clean on the live ledgers, the fourteen-pair replay unchanged.

Table 5. Dependency-invalidation replay: the synthetic twelve-fixture suite (unchanged) and the new fourteen-pair retroactive replay against real, git-verified commit-pairs in this project's own history.

Metric	Result
Synthetic fixture suite, exact match	12/12 (0 missed, 0 spurious)
Real-history replay, 14 commit-pairs	1,886/1,886 (0 missed, 0 spurious); precision 1.0, recall 1.0
Strongest real case (9-row re-slugging)	TP=9, FN=0, FP=0

6.4 Results: Historical Replay as Ecological Corroboration

Replaying the current checking machinery against six committed states of the ledger and corpus is ecological corroboration of prior, already-governed findings, not a controlled evaluation. Checker logic, extraction availability, corpus size, and status vocabulary all change together, so a rate change between two points cannot be attributed to one confound without separating them. One confound is separated: the witness-resolution repair's before/after count retired 0, 50, 75, and 77 spurious unresolved citations at the four snapshots, while extraction and intake volume moved independently. The resulting trajectory (Table 6) is non-monotone, dipping mid-series before recovering – directionally reproducing an already-published finding on an independent re-implementation, not a re-run of the original tool, which was never committed.

Table 6. Historical replay, current-checker machine-checkable rate against each snapshot's own commit-era corpus. Prior-published rates (final column) come from an earlier, differently-implemented replay and are not expected to match exactly; the direction – a dip mid-series, then recovery – is what this replay directionally reproduces, not exact-value corroboration.

Snapshot	Rows	This pass	Prior-published
Cycle 1	458	84.0%	93.2%
Session 7	682	81.5%	87.5%
Cycle 3	691	81.3%	87.2%
Cycle 4	697	94.4%	99.9%
This evaluation's snapshot	748	92.1%	—

A separate, full-history census replays every git-committed state of both ledgers – eleven quotation-ledger and ten claim-ledger states – finding one narrow VERIFIED-to-weaker reversal, four already-disclosed anomalous status cells at a single commit, and two commits with no status column at all, marked not-computable rather than zero change (Table 9).

6.5 Results: Claim-to-Quotation Coverage

A second ledger tracks argumentative claims against two conditions: the citability of the source a claim rests on, and, where the claim depends on a linked quotation, that quotation's status. As of 2026-08-24 – a live count outside the frozen-snapshot protocol above, unlike Tables 4–10 – the claim ledger carries 730 rows project-wide; 548 (75.1%) are linked to at least one quotation. Of those, 529 carry a single quotation link and resolve under a plain join – 489 from an exactly-checked quotation, 40 from a weaker text-verified one. The remaining 19 linked claims carry a semicolon-separated multi-value link; a generalized join resolves all 19 (16 strict, 3 broad) by the same rule. The remaining 182 claims carry no quotation link at all. This status vocabulary does not distinguish a claim inspected and left unresolved from one never inspected, a gap a nearby literature's status vocabulary names explicitly [Krebs 2026].

6.6 Results: Round 1 – Portability, Sensitivity, and Independent Agreement

Six further checks, run under a versioned addendum protocol (v1.1, frozen 2026-08-22), test whether the results reported above generalize past this corpus, this extraction pipeline, and the one implementation that produced them.

A disclosed OCR-sourced subpopulation – the checker's weakest point, narrowed from 29 candidate slugs to 8 confirmed-OCR ones (17 quotes) – re-verified at 88.2% VERIFIED, overlapping the rest of the corpus's 95.4% (754 rows), with both non-VERIFIED rows tracing to one already-disclosed OCR-corruption slug (Table 9). No comparable assessment exists for non-Latin-script or facsimile materials, neither present in this corpus.

The OQ-023-specified threshold-sensitivity sweep across five candidate thresholds moves zero of 771 rows to a different bucket under the recommended reclassification, closing OQ-023 with data – which threshold to adopt remains a policy choice, not a finding this check can settle (Table 9). That frozen 771-row pass disagreed with the ledger's stored status on 37 rows – fresh NOT_FOUND against a stored VERIFIED/TEXT-VERIFIED value, each listed individually rather than asserted as an error. A live

re-derivation and row-by-row adjudication – a different, 53-row disagreement set, not the 37 above – closes OQ-039 (Table A.1) with no miscited source found among them, though internally adjudicated.

An internal second-corpus transfer set – the paused MLA-track manuscript's 247-row quotation slice and 252-row claim slice, populated within the same project and tooling, drawn under the identical protocol – reproduced the catch-rate profile class for class, zero of five diverging: generalizing within this project's tooling, not to an independent implementation (Table 10).

Extraction-variant sensitivity, censused over every witness with more than one stored extraction, found 111 of 350 reachable rows (31.7%) verdict-sensitive to which variant the pipeline resolves – traced to extraction-method choice, not to de-hyphenation, which alone breaks or fixes zero of 267 rows (Table 10).

An independent second implementation, built primarily from the specification text with disclosed interface-level access, agreed with the deployed checker on 96.9% of 771 quotation verdicts (κ=0.665) and 97.8% of 715 claim verdicts (κ=0.954). Both disagreement sets trace to specification ambiguity, not an implementation defect (Table 10).

A PROV-O/PROV-N serialization of the full corpus (22,498 triples) passed all five scored PROV-Constraints checks – unique identifiers, no dangling reference, acyclic derivation, and two checks that pass trivially by construction (Table 10). It failed one disclosed interpretive proxy for generation-before-use timing on 42 of 83 scoreable pairs, tracing to one documented re-extraction sweep one calendar day off, over the 83 of 715 claim-quotation pairs that carried both timestamps needed to score it; the remaining 632 lacked one and were not scoreable. A live go-forward emission is reported in subsection A.2.

6.7 Results: A Direct Cross-Implementation Check and an Extraction-Sensitivity Follow-Up

Two further checks, run under protocol addendum v1.5, answer open questions above. A cross-implementation check ported the app's `findQuoteOffset` (`packages/claims/src/anchoring.ts`, read-only) against 771 frozen quote-ledger rows: agreement was 193/771 (25.0%, Wilson 95% CI [22.1, 28.2]%). Every disagreement traced to a disclosed gap (574 normalization, 4 elision), zero unexplained – a scope difference, with no claim about the live application or any database- or network-coupled predicate. Separately, 18 of the 111 extraction-sensitive rows above (16.2%) trace to this paper's own 98 cited keys, closing the claim above with a checked count, not an inference; which variant the pipeline resolved for those 18 remains unattempted.

6.8 No Semantic-Support Claim Is Made

Every check reported above answers whether a quoted string occurs in a held source, whether a citation resolves to the correct document, or whether a claim inherits a linked quotation's mechanical status – none tests whether the surrounding paraphrase is a faithful, entailed reading of what the quotation actually supports. This is the distinction the field's Attributable-to-Identified-Sources (AIS) framework draws between attribution and mere verbatim overlap [Rashkin et al. 2023]. A four-way annotation codebook for exactly that judgment (supported / contradicted / not-addressed / mixed) is specified and ready to run once qualified, independent human annotators are available; none is available this cycle, and no agent-generated label stands in for one. No semantic-support performance claim is made anywhere in this paper, for any claim–quotation pair, because no independently-labeled entailment judgment exists for any of them.

6.9 The Prior Self-Test Suite

Nine self-tests from earlier cycles remain the project's longest-running evidence base (Appendix B). Five are a provenance-table audit, a witness-ladder replay retiring sixty-nine spurious failures, a fault-injection sample reproduced at larger n in Table 4, a claim-provenance coverage census, and a repair-episode ablation. The remaining four are a historical replay extended by one point in Table 6, a review-trajectory analysis that caught a fabricated records claim, a reviewer-diversity analysis, and a silent-edit census; none is superseded here.

6.10 One Failure, Diagnosed

The most informative failure this checking machinery has produced did not start out as one: a quotation check flagged a real, previously verified source as failing to contain text a passage had quoted from it, and an adversarial review escalated it as a severe citation-integrity defect. The diagnosis was not a citation defect. The source is set in two columns, and the extractor had scanned across the page in one horizontal pass, interleaving unrelated lines into the compared string – the quoted text had sat in the source the whole time, never shown to the checker correctly assembled. The finding was withdrawn, becoming a standing rule: no not-found result is recorded until re-run against every plausible extraction variant – both de-hyphenation branches, both layout-preserving and reflowed settings – since some sources verify under only one. A parallel audit found the same defect in thirty-three of the then-manifested hundred and twenty sources, responsible for eighty-six false-negative checks and zero genuine hallucinations.

An automated integrity tool needing scrutiny of its own mirrors the statcheck debate in meta-science, where a re-derivation script produced both false-positive and false-negative findings from its own parsing limits [Nuijten et al. 2016; Schmidt 2016] – the nearest lineage, not identical. Table 11 (Appendix B) tabulates this episode alongside the three others diagnosed and repaired over the project's life. Thirty-five false-failure verdicts were retired total, none a quotation misrepresenting its source, and one weakness survived all four, disclosed rather than closed – a surname-and-year fallback rung that could still resolve an unmanifested conference paper against the wrong held work, unexercised on the current corpus.

7 Bounded Self-Application Case

The architecture above is our contribution: a verification-gated pipeline for AI-assisted research production, the object this paper evaluates. The reference implementation below illustrates that architecture and receives no evaluation section of its own, instancing the same unified method a second time rather than standing as a second system (§1). The project that produced this paper ran the architecture on its own production: one paper-writing agent built the argument above while a parallel build agent directed, across the same cycles, the construction of a reference implementation, studied as it currently stands, not as a finished artifact. Both operated under one human director. The two lines exchange findings but do not take each other's report on faith: a finding crossing from the build side is checked, read-only, against the build line's own source before the research side may use it. This runs at each cycle boundary and one-way, not a claim that the build line audits the research line's record in return. What follows is one bounded case of that discipline, a worked illustration, not independent validation across other teams, corpora, or domains.

One exchange shows the check running against the direction convenient to the paper: a draft proposed describing the reading plan's personalization as progressive withdrawal of support; the claim did not enter on the drafting agent's say-so. Checked, read-only, against the build line's own source, the finding came back narrower – five fixed stages, material re-ranked toward review as it is rated known, nothing removed and nothing new introduced as mastery grows. The paper reports the narrower finding because the wider one did not survive the check.

That check is the case's central point: a built capability is held unverified until read against its own source, mechanically and independent of who reports it – the same discipline the quotation ledger applies to a citation, extended from what a source says to what a system does. An agent distinct from the one that built a capability must verify it before either line may assert it as fact.

Three further capabilities pass the same discipline in brief. A credibility structure separates six independently displayed signals – publication rigor, creator expertise, host provenance, evidence strength, relevance, pedagogical value – rather than one confidence number, inheriting the same automation-bias exposure any such display carries [Kizilcec 2016; Pareek et al. 2024]. A retrieval layer excludes fabricated citations and claims lacking a grounded quotation, though the generated prose around a grounded citation is not checked sentence by sentence against what it says [Ji et al. 2023]. A discovery layer surfaces citation relations a reader never named, each carrying its own supporting quotation – relating a new claim to a source already held, not an unexplained assertion, the difference cognitive psychology calls self-explanation [Chi et al. 1981; Dunlosky et al. 2013; Kirschner et al. 2006]. Each capability entered the research record as a check the build line's own source passed, independent of whether it has been demonstrated in use.

One check is worth walking through in full, because it is the case's clearest instance of the architecture checking itself. An annotation layer reads a block of the primary text and asks a model to explain it; the build line enforces, mechanically, that an annotation's supporting quote must be a verbatim substring of the block it annotates, and drops it otherwise before it reaches a reader. Consider, as an illustration of that rule rather than a case drawn from the test suite, a block reading "Akrasia is weakness of will, acting against one's own better judgment." A candidate annotation quoting "weakness of will," a string the block actually contains, would be kept. One whose claimed quote does not occur in the block at all would be dropped – the same substring test the quotation ledger runs on a citation, run here on the tool's own reading of its own source. An annotation that survives that gate still reaches a reader as contestable rather than settled: the interface carries Verify, Dispute, and Reject controls on the same card. A reader who judges a genuinely grounded explanation wrong on its merits has a recorded way to say so – D5's reservation of interpretive judgment for the reader [Engelbart 1962; Licklider 1960], held at the point of contact, not merely inspectable after the fact. Whether a reader in practice uses that contest, or the card's transparency instead invites the deference Kizilcec and Pareek document, is left open; the design reserves the decision for the reader without showing that readers exercise it.

The case ran at project scale, not only in these two exchanges. The sibling manuscript's own corpus and ledger, replayed against four committed cycle closes (Table 11, Appendix B), show coverage moving from an adjusted 93.2% at the first cycle's close through a mid-project dip to 87.2%, recovering to 99.9% after a corpus-wide re-extraction audit. This is a non-monotone, repair-driven trajectory, with two cells left visibly in conflict rather than reconciled after the fact. The paper before the reader shows the same signature at draft scale: its own review chain, tracked round by round in Appendix B, moved through

repeated growth-then-repair cycles as review instruments changed and findings were closed, not a record of steady draft decay. None of this is a claim that the check generalizes past the one project, team, and corpus that ran it: self-application evidence licenses a claim about mechanism, never about generalization. Appendix B documents this case against that standard, row by row, and grows with each further cycle of the project's own process.

The project this paper reports applied its own architecture to verifying the artifact it was built to study, keeping the non-monotone record of that application, in committed form, rather than smoothing it after the fact. This case is offered in place of a deployment study, extended as each further review, test, or repair enters the project.

8 Discussion

Our claim is architectural: a citation that resolves against the held corpus or returns not-found is a mechanical, exact-match check – the predicate type §4's registry calls P1, not the weaker structural join a claim's inheritance gets or the interpretive judgment a gate review gives. It is decidable in that narrow sense, not a soft cue confidence can talk a reader out of noticing. Lowering the cost of verifying a claim, not adding more explanation, is the lever behavioral evidence supports for reducing overreliance – a mechanism-level warrant this design applies, not an outcome measured for any reader here [Vasconcelos et al. 2023]. That decidability has a measured referent: Lau et al.'s Critical Thinking in AI Use scale (N = 1,341) names "checking cited references" inside Verification, one of the construct's three constitutive factors. That factor allows verification to be initiated and supported by automatic processes through cognitive offloading [Lau et al. 2026] – a rational, metacognitively-governed strategy, not a capability deficit [Risko and Gilbert 2016]. What the architecture makes standing and low-cost is one behavior that factor's own definition names – a claim about when the behavior is available at no cost, not about any reader's disposition or score.

That gap invites a substitution objection: a script performs the check here, not the reader, so a resolved citation could as easily read as offloading the behavior as exercising it. D5 answers at the design level – whether a source may be trusted and whether an argument claims what it says stay reserved for the human [Engelbart 1962; Licklider 1960]. Every kept annotation carries Verify, Dispute, and Reject controls on its card (§7), requiring an explicit reader action, open to contest, rather than following automatically from the check having run – a design affordance, not a claim about what a reader in practice does with it. Bae et al.'s self-initiated-verification marker is that same factor's hedged behavioral referent, not evidence this architecture moves anyone's score [Bae et al. 2026]. A passing check settles fidelity, not calibrated trust: the reader should still match how much trust an LLM-mediated output earns to how trustworthy it actually is, a gap the check's own design leaves for the reader to close, not one it claims to close for them [Heersmink et al. 2024]. We also concede in full, rather than repeat here, the appropriate-reliance negative result and the Co-Scientist/Robin differentiation already stated in Related Work [Bo et al. 2025]. A preliminary DH editorial workspace under independent review reaches the same posture separately, treating a language model as an auditable assistant editors must check [Guérard et al. 2026].

A verification pipeline auditing its own research process counts as an HCI contribution the way mechanism-level evidence counts in place of interaction data for a systems-and-infrastructure contribution [Olsen 2007; Wobbrock and Kientz 2016]. The two lines this paper reports are one architecture instanced twice; the self-application makes that instancing visible rather than substituting for a reader study, and the architecture

presupposes, rather than replaces, a reader's competence. Judgment concentrates at the pipeline's human ratification points and the reader's own contest actions on generated annotations; a reader with no competency to exercise there is left with a checkable record whose use still depends on judgment the architecture cannot supply.

Internal validity. Every check reported here – on the implementation and this paper's own claims alike – was designed, run, and mostly reviewed by agents inside the one project, under the one director, that it audits: a common-mode risk no check here independently rules out. The checking machinery needed repair four times over the project's life, each diagnosed and converted into a standing rule rather than patched once – evidence the discipline catches its own faults, not that an outside party caught them. A first-party interview study finds HCI reviewers apply skeptical, inconsistent standards toward LLM-integrated systems papers, and that authors over-justify usage in response – a pressure this paper's own dense self-audit apparatus may itself be answering [Navarro et al. 2026]. What this kind of evidence licenses is a mechanism claim – this check ran, on this input, produced this output, inspectable by anyone who reads the record – never a generalization claim. Autobiographical- and research-through-design methods make that distinction explicit, holding outright that such work cannot produce results known to generalize to a broader community and carries no expectation that reproducing the process would reproduce the results [Neustaedter and Sengers 2012; Zimmerman et al. 2007]. A single self-applied case is confirmatory in shape, licensing the finding as genuinely checked, not fabricated, never that the method generalizes [Flyvbjerg 2006].

External validity. No claim is made, or evidenced, that this architecture transfers past the one project, one director, and one corpus that ran it – the research and build lines are not independent in a statistical or epistemic sense, sharing one director (§5). The fault-injection tests exercised five seeded corruption classes at $n = 30$ per class, later extended to $n = 100$/class with no class's threshold disposition changing (Table 4, §6) – not the wider space of compound corruptions the Coupling-Effect assumption in mutation testing takes on faith. Truncation is a destructive stress class, not a meaning-preserving control: a low catch rate there is expected, not a checker weakness. The held corpus is Latin-script, text-layer PDFs. A disclosed OCR-sourced subpopulation is separately measured at 88.2% VERIFIED (§6); facsimile-only sources and non-Latin scripts are neither present in the corpus nor assessed. The extraction-variant defect class §6 documents was diagnosed and repaired entirely inside that same Latin-script, text-layer scope. A Round-1 census of the current corpus quantified what remains: 31.7% of the reachable population still returns a different verdict by extraction variant alone (§6) – a limit distinct from, and larger than, the bug class the four repair episodes above closed.

Construct validity. "Decidable" and "mechanical" name three qualitatively different checks – exact, structural, and human-gated (§4–§6). No independently-labeled human judgment exists anywhere in this corpus for whether a cited quotation actually entails the claim built on it, a semantic-support gap the claim ledger's citability-and-status check does not close. A Round-1 reimplementation of the first two checks agreed with the deployed checker on the large majority of verdicts (§6). Its disagreements trace entirely to two named specification ambiguities, not an implementation bug on either side – evidence the specification is precise enough to reimplement, not that either reading is correct. A related, structural distinction (§6): the manuscript's own checking machinery still resweeps the entire ledger at each cycle boundary rather than selective, dependency-triggered re-verification [Mokhov et al. 2018] – kept because it caught two of the four checker repairs above, not because a selective mechanism was rejected. One level up, the pipeline's

record-keeping tooling implements that selective mechanism (§4, §6); the full resweep remains the backstop, unreplaced. LedgerMind's comparable staleness propagation is a property this pipeline's own tooling shares, differently scoped, not one it lacks [Du et al. 2026]. A related interface finds that provenance display can lower a reader's trust without lowering reliance on the system behind it [Martin-Boyle et al. 2026b] – a caution against assuming this architecture's own contestable-annotation card moves reliance as intended, since the card itself has not been tested against readers.

Future work should test, with independent readers who did not build or use this pipeline, whether the standing, low-cost availability of these checks measurably changes reliance calibration or erosion, rather than assuming the disposition literature's own hedge extends automatically to a designed occasion. No such reader trial has been run, and this paper claims none.

The review trail behind this paper is part of that same record, reported plainly rather than smoothed, diagrammed in full elsewhere (§5). The verifiable floor is severity-weighted: no submission-blocking finding is open, and every open item at the next severity sits in the records layer, not the argument. The same discipline holds across the pipeline's full review chain, documented step by step in the Self-Application Case (§5) and its process-evidence appendix, including at least one gate-caught failure disclosed there rather than smoothed away. A cell-level census of the two ledgers and the source manifest, over their full committed history, corroborates the same discipline at finer grain (§B.7). The standing AI-review stage answers the internal-validity gap above partway, putting the manuscript in front of a reviewer from a different model family, outside the pipeline's own tooling.

We also do not argue a further, constitutive question: whether a generated annotation counts as part of the machinery realizing a reader's own capacity, rather than a tool that capacity uses. No reader was measured here, and the extended-cognition literature above enters only where it does design work.

Ethics and anonymity. This paper reports no human-subjects data: no reader was recruited, observed, or measured, and the held corpus is licensed material its owner controls, not scraped or redistributed. The artifact package's licensing terms (`COPYRIGHT_NOTE.md`, `LICENSE`) govern redistribution of code and witness text; the architecture exposes no network-facing API, only local tooling over this closed corpus.

This manuscript was anonymized per venue policy during peer review, with no identifying detail exposed outside the sanctioned AI-disclosure statement; this public copy restores that information. The data-provenance appendix (Appendix A) traces every project-process figure to a described project record rather than to a literal filename, commit identifier, or other searchable internal marker.

9 Conclusion

This paper describes one unified method, not two adjacent projects. Two parallel moderating agents carried it out: one conducted the research and drafting reported here; the other built, from the same design desiderata, the reference implementation §7 describes. Each line answers to the same gates, ledgers, and human ratification points, and each depends on the other: the research line's own verification discipline instances the architecture it argues for, and the implementation is the object that discipline holds accountable. The project is itself a second, load-bearing case of that same pattern, alongside the reference implementation it describes. Whether the pattern generalizes past this one instance is an untested conjecture the design supports, not a demonstrated transfer. The conjecture holds two conditions together. First, wherever a corpus can be held, a citation or quotation check can be made mechanically decidable in the narrow, exact-match

sense argued above – not claim-inheritance or gate review, which stay structural and interpretive respectively. Second, a human is willing to close the loop with judgment rather than confidence. Where both hold, the index–verify–gate pattern – grounded here in the humanities' own citation-practice problem (§4) – is available, we conjecture, to another human-directed, corpus-bounded research program.

Acknowledgments

Generative language models supported the moderating agents throughout this pipeline: literature search, drafting and revision of prose under human-set briefs, and the mechanical and adversarial checks above. Large-language-model assistants (two vendors) were used. One assistant supported the pipeline's own agents, across three capability tiers by task type – routine and mechanical work, literature synthesis and drafting, and moderator-level verification judgments. A second, different-vendor assistant served as an external reviewer for one independent pre-submission audit, outside the pipeline's own tooling. The human director set every research and framing decision, ratified every gate the project's own record shows ratified, and bears responsibility for this paper. All listed authors, not the director alone, are accountable for this paper's content, including any AI-assisted portions, regardless of source, per ACM's authorship policy. No generative model is listed as an author.

Data availability. A supplementary artifact package accompanies this paper, containing the predicate registry, the evaluation protocol, the frozen evaluation inputs, the analysis scripts, and the run outputs. The package targets the *Artifacts Available* standard [Association for Computing Machinery 2020]: no party independent of the authors has reproduced its results as of this writing. The independent-re-implementation comparison is closer to that standard's *Replicated* sense – a second implementation from the same specification, not a shared artifact – than to *Reproduced*, which uses the authors' own artifact [Association for Computing Machinery 2020]. The package is dual-licensed, per its own `LICENSE` file: the reproduction scripts under the MIT License, and the ledger, manifest, and documentation excerpts under CC BY 4.0. Two categories of material are deliberately excluded (disclosed in the package's limitations document): the extracted-text corpus the quotation-verification check reads against, withheld as derived, near-full-text extraction from copyrighted third-party academic sources. The other is the project's full version-control history, read by one evaluation component and not partially includable. Where a component depends on either excluded resource, the package ships that component's already-executed output, the script that produced it, and a stated account of what an independent reviewer can and cannot confirm from the package. One component, the dependency-invalidation-replay fixture suite, is self-contained and reproduces its committed output byte-for-byte from the package's contents.

On accessibility: this PDF declares its language. Figure alt text remains blocked: every figure carries a `\Description{}` argument in the source, but `acmart`'s own `Description` macro discards that argument before it reaches the compiled PDF, a class-level limitation this build toolchain cannot close. Table header-cell tagging was investigated and found achievable, not blocked: enabling `acmart`'s `\DocumentMetadata`-driven tagged-PDF path (`tagpdf`, `latex-lab`, `pdfmanagement`) tags all nineteen of the v16 build's tables' header cells correctly. The paper's table set has changed since; no table added, removed, or altered after that build has been retested. A four-pass rebuild under that path produced zero layout regression against the untagged build – an identical page count and an identical Overfull-hbox set. Rather than integrate that

toolchain change under deadline pressure, tagging is instead applied to the submission PDF by the authors: a submission-time Acrobat Pro auto-tag pass closes the table-tagging gap without a build-pipeline change. It does not solve the figure-tagging problem, which remains blocked regardless of tagging path.

This paper's supplementary artifact package, described above under Data availability, is released alongside it for readers who wish to inspect the underlying ledgers, scripts, and run outputs.

References

ACM SIGCHI. 2026. Contributions to CHI. https://chi2027.acm.org/contributions-to-chi/ Accessed: 2026-08-22.

Krishna Akhil Kumar Adavi, Pratik Ghosh, Richard Banks, Advait Sarkar, and Sian E. Lindley. 2026. "Will This Tool Ever Push Back or Challenge Me?": Reflections on a Multi-agent LLM Tool for Perspective Seeking. In *Proceedings of the 5th Annual Symposium on Human-Computer Interaction for Work* (Linz, Austria) *(CHIWORK '26)*. Association for Computing Machinery, New York, NY, USA, 16 pages. doi:10.1145/3808045.3808058

Gabriel Amaral, Odinaldo Rodrigues, and Elena Simperl. 2024. ProVe: A pipeline for automated provenance verification of knowledge graphs against textual sources. *Semantic Web* 15 (2024). doi:10.3233/SW-233467

Saleema Amershi, Dan Weld, Mihaela Vorvoreanu, Adam Fourney, Besmira Nushi, Penny Collisson, Jina Suh, Shamsi Iqbal, Paul N. Bennett, Kori Inkpen, Jaime Teevan, Ruth Kikin-Gil, and Eric Horvitz. 2019. Guidelines for Human-AI Interaction. In *Proceedings of the 2019 CHI Conference on Human Factors in Computing Systems*. 1–13. doi:10.1145/3290605.3300233

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2024. SELF-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. *arXiv preprint arXiv:2310.11511* (2024). doi:10.48550/arXiv.2310.11511

Association for Computing Machinery. 2020. Artifact Review and Badging Version 1.1. ACM Publications Policies. https://www.acm.org/publications/policies/artifact-review-and-badging-current

Seunghyun Bae, Hyungwoo Song, Hyeon Jeon, Taesup Kim, and Bongwon Suh. 2026. Supporting or Hindering? Understanding the Divergent Effects of LLM Assistance on Critical Thinking in Academic Reading. In *Extended Abstracts of the 2026 CHI Conference on Human Factors in Computing Systems* (Barcelona, Spain) *(CHI EA '26)*. Association for Computing Machinery, New York, NY, USA, 5 pages. doi:10.1145/3772363.3798788

Hamsa Bastani, Osbert Bastani, Alp Sungu, Haosen Ge, Özge Kabakcı, and Rei Mariman. 2025. Generative AI without Guardrails Can Harm Learning: Evidence from High School Mathematics. *Proceedings of the National Academy of Sciences* 122, 26, Article e2422633122 (2025). doi:10.1073/pnas.2422633122

David M. Berry. 2025. AI Sprints: Towards a Critical Method for Human-AI Collaboration. *arXiv preprint arXiv:2512.12371* (2025). doi:10.48550/arXiv.2512.12371

Jessica Y. Bo, Sophia Wan, and Ashton Anderson. 2025. To Rely or Not to Rely? Evaluating Interventions for Appropriate Reliance on Large Language Models. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '25)*. 24 pages. doi:10.1145/3706598.3714097

Lawrence D. Brown, T. Tony Cai, and Anirban DasGupta. 2001. Interval Estimation for a Binomial Proportion. *Statist. Sci.* 16, 2 (2001), 101–133. doi:10.1214/ss/1009213286

Zana Buçinca, Maja Barbara Malaya, and Krzysztof Z. Gajos. 2021. To Trust or to Think: Cognitive Forcing Functions Can Reduce Overreliance on AI in AI-assisted Decision-making. *Proceedings of the ACM on Human-Computer Interaction* 5, CSCW1, Article 188 (2021). doi:10.1145/3449287

Gengyu Chen, Yongjie Yu, and Weiling Wang. 2026. Evidence-Ledger Adjudication for Claim-Evidence Traceability. arXiv:2607.26512 [cs.CL]

Michelene T. H. Chi, Paul J. Feltovich, and Robert Glaser. 1981. Categorization and Representation of Physics Problems by Experts and Novices. *Cognitive Science* 5, 2 (1981), 121–152. doi:10.1207/s15516709cog0502_2

Andy Clark. 2015. What 'Extended Me' Knows. *Synthese* 192 (2015), 3757–3775. doi:10.1007/s11229-015-0719-z

Andy Clark. 2025. Extending Minds with Generative AI. *Nature Communications* 16, Article 4627 (2025). doi:10.1038/s41467-025-59906-9

Tim Clark, Paolo N. Ciccarese, and Carole A. Goble. 2014. Micropublications: a semantic model for claims, evidence, arguments and annotations in biomedical communications. Journal of Biomedical Semantics 5:28 (2014). doi:10.1186/2041-1480-5-28

Audrey Desjardins and Aubree Ball. 2018. Revealing Tensions in Autobiographical Design in HCI. In *Proceedings of the 2018 Designing Interactive Systems Conference (DIS 2018)*. Hong Kong, 753–764. doi:10.1145/3196709.3196781

Ian Drosos, Advait Sarkar, Xiaotong (Tone) Xu, and Neil Toronto. 2025. "It Makes You Think": Provocations Help Restore Critical Thinking to AI-Assisted Knowledge Work. *arXiv preprint arXiv:2501.17247* (2025). doi:10.48550/arXiv.2501.17247

Enjun Du, Hange Zhou, Chenxu Du, Siyi Liu, Zirong Chen, Ziyu Zheng, and Yongqi Zhang. 2026. LedgerMind: A Structured Evidence Runtime for Auditable Multimodal Agent Trajectories. arXiv:2607.28374 [cs.LG]

John Dunlosky, Katherine A. Rawson, Elizabeth J. Marsh, Mitchell J. Nathan, and Daniel T. Willingham. 2013. Improving Students' Learning with Effective Learning Techniques: Promising Directions from Cognitive and Educational Psychology. *Psychological Science in the Public Interest* 14, 1 (2013), 4–58. doi:10.1177/1529100612453266

Douglas C. Engelbart. 1962. *Augmenting Human Intellect: A Conceptual Framework*. Technical Report AFOSR-3223. Stanford Research Institute.

Xinrui Fang, Anran Xu, Chi-Lan Yang, Ya-Fang Lin, Sylvain Malacria, and Koji Yatani. 2026. LLM-based In-situ Thought Exchanges for Critical Paper Reading. In *31st International Conference on Intelligent User Interfaces* (Paphos, Cyprus) *(IUI '26)*. 22 pages. doi:10.1145/3742413.3789069

Bent Flyvbjerg. 2006. Five Misunderstandings About Case-Study Research. *Qualitative Inquiry* 12, 2 (2006), 219–245. doi:10.1177/1077800405284363

Selina Galka and Georg Vogeler. 2026. Annotating, Projecting, and Interpreting Named Entities in Digital Scholarly Editions with LLMs. *International Journal of Digital Humanities* (2026). doi:10.1007/s42803-025-00114-8

Luyu Gao, Zhuyun Dai, Panupong Pasupat, Anthony Chen, Arun Tejasvi Chaganty, Yicheng Fan, Vincent Y. Zhao, Ni Lao, Hongrae Lee, Da-Cheng Juan, and Kelvin Guu. 2023a. RARR: Researching and Revising What Language Models Say, Using Language Models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. doi:10.18653/v1/2023.acl-long.910

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. 2023b. Enabling Large Language Models to Generate Text with Citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP 2023)* (Singapore). Association for Computational Linguistics, 6465–6488. doi:10.18653/v1/2023.emnlp-main.398

Bill Gaver and John Bowers. 2012. Annotated Portfolios. *Interactions* 19, 4 (2012), 40–49. doi:10.1145/2212877.2212889

Michael Gerlich. 2025. AI Tools in Society: Impacts on Cognitive Offloading and the Future of Critical Thinking. *Societies* 15, 1, Article 6 (2025). doi:10.3390/soc15010006

Ali E. Ghareeb, Benjamin Chang, Ludovico Mitchener, Angela Yiu, Caralyn J. Szostkiewicz, Dmytro Shved, Gavin J. Gyimesi, Jon M. Laurent, Samantha M. Wright, Muhammed T. Razzak, Andrew D. White, Silvia C. Finnemann, Michaela M. Hinks, and Samuel G. Rodriques. 2026. A Multi-agent System for Automating Scientific Discovery. *Nature* 655 (2026), 497–505. doi:10.1038/s41586-026-10652-y

Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Petar Sirkovic, Artiom Myaskovsky, Grzegorz Glowaty, Felix Weissenberger, Alessio Orlandi, Dan Popovici, Anil Palepu, Keran Rong, Ryutaro Tanno, Khaled Saab, Fan Zhang, Jacob Blum, Andrew Carroll, Kavita Kulkarni, Nenad Tomasev, Dina Zverinski, Ivor Rendulic, Elahe Vedadi, Florian Hasler, Luka Rimanic, Marina Boia, Ivan Budiselic, Ben Feinstein, Mathias Bellaiche, Tom Sheffer, Jan Freyberg, Jeremy Ratcliff, Ottavia Bertolli, Katherine Chou, Avinatan Hassidim, Burak Gokturk, Amin Vahdat, Yuan Guan, Vikram Dhillon, Eeshit Dhaval Vaishnav, Byron Lee, Tiago R. D. Costa, Jose R. Penades, Gary Peltz, Yossi Matias, James Manyika, Demis Hassabis, Yunhan Xu, Pushmeet Kohli, Annalisa Pawlosky, Alan Karthikesalingam, and Vivek Natarajan. 2026. Accelerating Scientific Discovery with Co-Scientist. *Nature* 655 (2026), 487–496. doi:10.1038/s41586-026-10644-y

Paul Groth, Andrew Gibson, and Jan Velterop. 2010. The Anatomy of a Nanopublication. *Information Services & Use* 30, 1-2 (2010), 51–56. doi:10.3233/ISU-2010-0613

Élisabeth Guérard, Mehrdad Almasi, Marion Salaün, Frédéric Clavert, and Mirjam Pfeiffer. 2026. Towards an Interactive Evidence-RAG Peer-Review Workspace for the Journal of Digital History. arXiv:2606.25837 [cs.DL]

Sebastian Haan. 2025. SemanticCite: Citation Verification with AI-Powered Full-Text Analysis and Evidence-Based Reasoning. *arXiv preprint arXiv:2511.16198* (2025). doi:10.48550/arXiv.2511.16198

R. D. Hawkins and T. P. Kelly. 2012. A Framework for Determining the Sufficiency of Software Safety Assurance. In *7th IET International Conference on System Safety, incorporating the Cyber Security Conference 2012*. Institution of Engineering and Technology, 61. doi:10.1049/cp.2012.1529

Richard Heersmink, Barend de Rooij, Jimena Clavel Vázquez, and Matteo Colombo. 2024. A Phenomenology and Epistemology of Large Language Models: Transparency, Trust, and Trustworthiness. *Ethics and Information Technology* 26, 3 (2024), 41. doi:10.1007/s10676-024-09777-3

Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Yejin Bang, Delong Chen, Wenliang Dai, Ho Shu Chan, Andrea Madotto, and Pascale Fung. 2023. Survey of Hallucination in Natural Language Generation. *Comput. Surveys* 55, 12, Article 248 (2023). doi:10.1145/3571730

Deyu Jing. 2026. Traceable Scholarship: Page Anchors and Ariadne's Thread for Humanistic Inquiry in the Age of Generative AI. arXiv:2607.20916 [cs.CL]

Daehong Kim, Haichao Miao, and Shusen Liu. 2026. LEDGER: Claim-to-Evidence Trace Graphs for Auditing LLM Agents. arXiv:2608.18398 [cs.CL]

Paul A. Kirschner, John Sweller, and Richard E. Clark. 2006. Why Minimal Guidance During Instruction Does Not Work: An Analysis of the Failure of Constructivist, Discovery, Problem-Based, Experiential, and Inquiry-Based Teaching. *Educational Psychologist* 41, 2 (2006), 75–86. doi:10.1207/s15326985ep4102_1

David Kirsh and Paul Maglio. 1994. On Distinguishing Epistemic from Pragmatic Action. *Cognitive Science* 18, 4 (1994), 513–549. doi:10.1207/s15516709cog1804_1

René F. Kizilcec. 2016. How Much Information?: Effects of Transparency on Trust in an Algorithmic Interface. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems* (San Jose, CA, USA). 2390–2395. doi:10.1145/2858036.2858402

Lauren Klein, Meredith Martin, André Brock, Maria Antoniak, Melanie Walsh, Jessica Marie Johnson, Lauren Tilton, and David Mimno. 2025. Provocations from the Humanities for Generative AI Research. *arXiv preprint arXiv:2502.19190* (2025). doi:10.48550/arXiv.2502.19190

Guneet Kohli. 2026. Nine Judges, Two Effective Votes: Correlated Errors Undermine LLM Evaluation Panels. *arXiv preprint arXiv:2605.29800* (2026). doi:10.48550/arXiv.2605.29800

Nataliya Kosmyna, Eugene Hauptmann, Ye Tong Yuan, Jessica Situ, Xian-Hao Liao, Ashly Vivian Beresnitzky, Iris Braunstein, and Pattie Maes. 2025. Your Brain on ChatGPT: Accumulation of Cognitive Debt when Using an AI Assistant for Essay Writing Task. *arXiv preprint arXiv:2506.08872* (2025). doi:10.48550/arXiv.2506.08872

Florian Krebs. 2026. F(AI)2R: Who Did What, and Who Checked? Verifiable AI Provenance as an Executable Skill. arXiv:2607.25637 [cs.CL]

Gabriel R. Lau, Wei Yan Low, Louis Tay, Ysabel A. Guevarra, Dragan Gašević, and Andree Hartanto. 2026. Understanding Critical Thinking in Generative Artificial Intelligence Use: Development, Validation, and Correlates of the Critical Thinking in AI Use Scale. *Computers in Human Behavior Reports* 22, Article 101103 (2026). doi:10.1016/j.chbr.2026.101103

Timothy Lebo, Satya Sahoo, and Deborah McGuinness. 2013. *PROV-O: The PROV Ontology*. Technical Report. W3C Recommendation. https://www.w3.org/TR/

Hao-Ping Lee, Advait Sarkar, Lev Tankelevitch, Ian Drosos, Sean Rintel, Richard Banks, and Nicholas Wilson. 2025. The Impact of Generative AI on Critical Thinking: Self-Reported Reductions in Cognitive Effort and Confidence Effects From a Survey of Knowledge Workers. In *Proceedings of the 2025 CHI Conference on Human Factors in Computing Systems* (Yokohama, Japan) *(CHI '25)*. Association for Computing Machinery, New York, NY, USA, 22 pages. doi:10.1145/3706598.3713778

John D. Lee and Katrina A. See. 2004. Trust in Automation: Designing for Appropriate Reliance. *Human Factors: The Journal of the Human Factors and Ergonomics Society* 46, 1 (2004), 50–80. doi:10.1518/hfes.46.1.50_30392

J. C. R. Licklider. 1960. Man-Computer Symbiosis. *IRE Transactions on Human Factors in Electronics* HFE-1 (1960), 4–11. doi:10.1109/THFE2.1960.4503259

Sophia Liu, Sarah Abowitz, Yijun Liu, Sarah Sterman, Shm Garanganao Almeda, and Max Kreminski. 2026. Creative Reading: Scaffolding Reading for Transformation. In *37th ACM Conference on Hypertext (HT '26)* (London, United Kingdom) *(HT '26)*. Association for Computing Machinery, New York, NY, USA, 8 pages. doi:10.1145/3800935.3830891 Forthcoming; conference scheduled September 14–18, 2026, after this manuscript's own completion date..

Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. 2024. The AI Scientist: Towards Fully Automated Open-Ended Scientific Discovery. *arXiv preprint arXiv:2408.06292* (2024). doi:10.48550/arXiv.2408.06292

Anna Martin-Boyle, William Humphreys, Martha Brown, Cara Leckey, and Harmanpreet Kaur. 2026a. An Expert Schema for Evaluating Large Language Model Errors in Scholarly Question-Answering Systems. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems (CHI '26)*. doi:10.1145/3772318.3791843

Anna Martin-Boyle, Cara A. C. Leckey, Martha C. Brown, and Harmanpreet Kaur. 2026b. PaperTrail: A Claim-Evidence Interface for Grounding Provenance in LLM-based Scholarly Q&A. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems (CHI '26)*. doi:10.1145/3772318.3791101

Rui Meng, Bhavana Dalvi Mishra, Jiefeng Chen, Chun-Liang Li, Palash Goyal, Mihir Parmar, Yiwen Song, Yale Song, Rajarishi Sinha, Parthasarathy Ranganathan, Burak Gokturk, Jinsung Yoon, and Tomas Pfister. 2026. ScientistOne: Towards Human-Level Autonomous Research via Chain-of-Evidence. arXiv:2605.26340 [cs.CL]

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2023. FActScore: Fine-grained Atomic Evaluation of Factual Precision in Long Form Text Generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. doi:10.18653/v1/2023.emnlp-main.741

Andrey Mokhov, Neil Mitchell, and Simon Peyton Jones. 2018. Build Systems a la Carte. In *Proceedings of the ACM on Programming Languages, Volume 2, Issue ICFP, Article 79*. doi:10.1145/3236774

Luc Moreau and Paolo Missier. 2013a. *PROV-DM: The PROV Data Model*. Technical Report. W3C Recommendation. https://www.w3.org/TR/

Luc Moreau and Paolo Missier. 2013b. *PROV-N: The Provenance Notation*. Technical Report. W3C Recommendation. https://www.w3.org/TR/

Karla Felix Navarro, Eugene Syriani, and Ian Arawjo. 2026. Reporting and Reviewing LLM-Integrated Systems in HCI: Challenges and Considerations. arXiv:2602.05128 [cs.HC]

Carman Neustaedter and Phoebe Sengers. 2012. Autobiographical Design in HCI Research: Designing and Learning through Use-It-Yourself. In *Proceedings of the Designing Interactive Systems Conference (DIS 2012)*. Newcastle, UK, 514–523.

M. B. Nuijten, C. H. J. Hartgerink, M. A. L. M. van Assen, S. Epskamp, and J. M. Wicherts. 2016. The Prevalence of Statistical Reporting Errors in Psychology (1985–2013). *Behavior Research Methods* 48, 4 (2016), 1205–1226. doi:10.3758/s13428-015-0664-2

Dan R. Olsen, Jr. 2007. Evaluating User Interface Systems Research. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology (UIST '07)*. doi:10.1145/1294211.1294256

Yating Pan, Jiajun Zhang, Jun Wang, and Qi Su. 2026. Extending AI for Research to the Humanities: A Multi-Agent Framework for Evidence-Grounded Scholarship. *arXiv preprint arXiv:2605.30947* (2026). doi:10.48550/arXiv.2605.30947

Mike Papadakis, Marinos Kintis, Jie Zhang, Yue Jia, Yves Le Traon, and Mark Harman. 2019. Mutation Testing Advances: An Analysis and Survey. Advances in Computers, Volume 112 (Elsevier academic book series), pp. 275-378.

Saumya Pareek, Niels van Berkel, Eduardo Velloso, and Jorge Goncalves. 2024. Effect of Explanation Conceptualisations on Trust in AI-assisted Credibility Assessment. *Proceedings of the ACM on Human-Computer Interaction* 8, CSCW2, Article 383 (2024). doi:10.1145/3686922

Samir Passi and Mihaela Vorvoreanu. 2022. *Overreliance on AI: Literature Review*. Technical Report. Microsoft Aether (AI Ethics and Effects in Engineering and Research).

Hannah Rashkin, Vitaly Nikolaev, Matthew Lamm, Lora Aroyo, Michael Collins, Dipanjan Das, Slav Petrov, Gaurav Singh Tomar, Iulia Turc, and David Reitter. 2023. Measuring Attribution in Natural Language Generation Models. *Computational Linguistics* 49, 4 (2023), 777–840. doi:10.1162/coli_a_00486

Patrik Reizinger and Wieland Brendel. 2026. HALLMARK: Diagnosing Three Failure Modes in LLM Citation Verifiers. *arXiv preprint arXiv:2607.18360* (2026). doi:10.48550/arXiv.2607.18360

Evan F. Risko and Sam J. Gilbert. 2016. Cognitive Offloading. *Trends in Cognitive Sciences* 20, 9 (2016), 676–688. doi:10.1016/j.tics.2016.07.002

Robert Sanderson, Paolo Ciccarese, and Benjamin Young. 2017. *Web Annotation Data Model*. Technical Report. W3C Recommendation. https://www.w3.org/TR/

Michael Schlichtkrull, Zhijiang Guo, and Andreas Vlachos. 2023. AVeriTeC: A Dataset for Real-world Claim Verification with Evidence from the Web. In *NeurIPS 2023 (37th Conference on Neural Information Processing Systems), Datasets and Benchmarks Track*.

T. Schmidt. 2016. Sources of False Positives and False Negatives in the STATCHECK Algorithm: Reply to Nuijten et al. (2016). arXiv:1610.01010. https://arxiv.org/abs/1610.01010

Edilson Silva, Edgar Tanaka, and Gustavo Tordin. 2019. Dogfooding: "Eating Our Own Dog Food" in a Large Global Mobile Industry Player. In *2019 ACM/IEEE 14th International Conference on Global Software Engineering (ICGSE)*. IEEE, 52–57. doi:10.1109/ICGSE.2019.00024

Paul R. Smart. 2022. Toward a Mechanistic Account of Extended Cognition. *Philosophical Psychology* 35, 8 (2022), 1107–1135. doi:10.1080/09515089.2021.2023123

Paul R. Smart. 2024. Extended X: Extending the Reach of Active Externalism. *Cognitive Systems Research* 84, Article 101202 (2024). doi:10.1016/j.cogsys.2023.101202

Renan Souza, Amal Gueroudji, Stephen DeWitt, Daniel Rosendo, Tirthankar Ghosal, Robert Ross, Prasanna Balaprakash, and Rafael Ferreira da Silva. 2025. PROV-AGENT: Unified Provenance for Tracking AI Agent Interactions in Agentic Workflows. In *Proceedings of the 21st IEEE International Conference on e-Science (e-Science 2025)*. arXiv:2508.02866 [cs.DC]

Milos Stankovic, Ella Hirche, Sarah Kollatzsch, and Julia Nadine Doetsch. 2026. Commentary on Kosmyna, N., Hauptmann, E., Yuan, Y. T., Situ, J., Liao, X. H., Beresnitzky, A. V., ... & Maes, P. (2025). Your Brain on ChatGPT: Accumulation of Cognitive Debt when Using an AI Assistant for Essay Writing Task. *arXiv preprint arXiv:2601.00856* (2026). doi:10.48550/arXiv.2601.00856

Kyle Swanson, Wesley Wu, Nash L. Bulaong, John E. Pak, and James Zou. 2025. The Virtual Lab of AI Agents Designs New SARS-CoV-2 Nanobodies. *Nature* 646 (2025), 716–723. doi:10.1038/s41586-025-09442-9

John Sweller. 1994. Cognitive Load Theory, Learning Difficulty, and Instructional Design. *Learning and Instruction* (1994). doi:10.1016/0959-4752(94)90003-5

Stefan Szeider. 2026. Unmediated AI-Assisted Scholarly Citations. In *The Second Bridge on Artificial Intelligence for Scholarly Communication (AAAI-26)*. doi:10.52825/ocp.v8i.3161

Neşet Özkan Tan, Minghao Li, and Mark Gahegan. 2026. SciTrue: Evidence-Grounded Scientific Claim Verification and Traceable Summarization. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2026), Volume 3: System Demonstrations*. doi:10.18653/v1/2026.eacl-demo.27

Lev Tankelevitch, Viktor Kewenig, Auste Simkute, Ava Elizabeth Scott, Advait Sarkar, Abigail Sellen, and Sean Rintel. 2024. The Metacognitive Demands and Opportunities of Generative AI. In *Proceedings of the CHI Conference on Human Factors in Computing Systems (CHI '24)* (Honolulu, HI, USA) *(CHI '24)*. Association for Computing Machinery, New

York, NY, USA, 24 pages. doi:10.1145/3613904.3642902

Harry Yizhou Tian, Hasan Amin, and Ming Yin. 2026. Understanding the Effects of AI-Assisted Critical Thinking on Human-AI Decision Making. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems (CHI '26)*. doi:10.1145/3772318.3790785

John Unsworth. 2000. Scholarly Primitives: What Methods Do Humanities Researchers Have in Common, and How Might Our Tools Reflect This? https://www.iath.virginia.edu/~jmu2m/primitives.html Invited talk, symposium on "Humanities Computing: formal methods, experimental practice," King's College London, 13 May 2000.

Helena Vasconcelos, Matthew Jörke, Madeleine Grunde-McLaughlin, Tobias Gerstenberg, Michael S. Bernstein, and Ranjay Krishna. 2023. Explanations Can Reduce Overreliance on AI Systems During Decision-Making. In *Proceedings of the ACM on Human-Computer Interaction, Volume 7, Issue CSCW1, Article 129*. doi:10.1145/3579605

Mireia Vendrell and Samantha-Kaye Johnston. 2026. Scaffolding Critical Thinking with Generative AI: Design Principles for Integrating Large Language Models in Higher Education. *Computers and Education: Artificial Intelligence* 10, Article 100572 (2026). doi:10.1016/j.caeai.2026.100572

Apurv Verma. 2026. Is this Citation on Point? arXiv:2608.12571 [cs.DL]

David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. 2020. Fact or Fiction: Verifying Scientific Claims. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. doi:10.18653/v1/2020.emnlp-main.609

Yiqi Wang, Jiaqi Zhang, Zhangkai Wu, Taotao Cai, Zirui Liu, Qingqiang Sun, Zequn Sun, Manqing Dong, Mingkai Zheng, Xuefei Yin, and Yanming Zhu. 2026. From Agent Traces to Trust: A Survey of Evidence Tracing and Execution Provenance in LLM Agents. arXiv:2606.04990 [cs.CR]

Edwin B. Wilson. 1927. Probable Inference, the Law of Succession, and Statistical Inference. *J. Amer. Statist. Assoc.* 22, 158 (1927), 209–212. doi:10.1080/01621459.1927.10502953

Jacob O. Wobbrock and Julie A. Kientz. 2016. Research Contributions in Human-Computer Interaction. ACM Interactions 23(3). doi:10.1145/2907069

Yutaro Yamada, Robert Tjarko Lange, Cong Lu, Shengran Hu, Chris Lu, Jakob Foerster, Jeff Clune, and David Ha. 2025. The AI Scientist-v2: Workshop-Level Automated Scientific Discovery via Agentic Tree Search. *arXiv preprint arXiv:2504.08066* (2025). doi:10.48550/arXiv.2504.08066

Jiayin Zhi, Harsh Kumar, and Mina Lee. 2026a. Investigating the Effects of LLM Use on Critical Thinking under Time Constraints: Access Timing and Time Availability. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems* (Barcelona, Spain) *(CHI '26)*. 21 pages. doi:10.1145/3772318.3791796

Jiayin Zhi, Hoyt Long, Richard Jean So, and Mina Lee. 2026b. What Does AI Do for Cultural Interpretation? A Randomized Experiment on Close Reading Poems with Exposure to AI Interpretation. In *Proceedings of the 2026 CHI Conference on Human Factors in Computing Systems* (Barcelona, Spain) *(CHI '26)*. 18 pages. doi:10.1145/3772318.3791727

Jing Zhou, Li Si, and Wenjun Hou. 2025. Humanities-in-the-Loop: Using Close Reading as a Method for Retrieval-Augmented Generation (RAG). *Proceedings of the Association for Information Science and Technology* 62, 1 (2025), 1747–1749. doi:10.1002/pra2.1529

John Zimmerman, Jodi Forlizzi, and Shelley Evenson. 2007. Research Through Design as a Method for Interaction Design Research in HCI. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI 2007)*. San Jose, CA, USA, 493–502.

A Data-Provenance Table

Artifact pointer. Per author ruling USER-113 (2026-08-23, task R7.W-APPENDIX), this appendix and Appendix B keep only a compressed provenance table, a compressed self-application record, and the review-round findings figure; every table compressed since USER-113 is reproduced in full, verbatim, in the anonymized supplementary package (CSV and Markdown twins) as Tables S1–S16. Every headline figure those tables support is still stated directly in the body prose of §5–§7; only the full per-row detail moved.

Non-load-bearing note. This appendix and the supplementary package (Appendix B; Tables S1–S16) hold depth and audit trail, not evidence a reader needs to reach to evaluate this paper: every headline figure they support is independently restated in §5–§7's own body prose, confirmed sentence-by-sentence at each restructuring pass (most recently 2026-08-23). A reader who consults only this paper's own body sections is not missing a load-bearing number.

Every project-process figure asserted in §5–§7 traces to the record it was read from here, so a third party holding the repository can recount each one. All values are point-in-time snapshots, re-derived at each round's own pass (most recently 2026-08-23), not from a single fixed final build. Ledgers grow during active intake, so a later reader must recount against the project's own records at their own read time; an earlier run of this audit caught four counters one intake batch stale, re-pinned rather than reconciled silently. Each locator pin below is checked by a committed audit script: it confirms every pin's lines exist, stay in bounds, and never fall on a blank or comment-only line, and that a clause closes at or before the pin's start and within its span. A reader can re-run it rather than take a pin on trust. This paper's $\leq$50-word sentence-length convention governs body prose only, not table cells, captions, or bibliography entries.

Both tables below (Table 7, Table 8) were compressed in stages to recover appendix space. The first stage moved from one row per individual claim (34 rows, 34 pins) to one row per claim group (11 rows, 21 pins) – the version reproduced in full as Tables S15–S16 in the supplementary package. This pass compresses the printed appendix further, to three exemplar rows per table, with the total claim-group and claim counts stated in each table's own caption and the remaining groups' figures reproduced only in the supplement. Two individual claims a claim-group row once counted – a handoff-specification detail and a review-prompt line count – were genuinely cut from §5's prose by an earlier trim with no replacement clause, and are removed from the claim list rather than pinned to text that no longer makes them.

Two prose figures are deliberately absent from either table: the nine-judges/seven-families statistic and its two-votes corollary (§5) are a cited literature finding, not a project record, and §7's akrasia walk-through is labeled an illustration in the text itself. The cycle-close summary table and the coverage-trajectory figure that §7 cites are traced here by their headline series only. Individual pages and terminal-review cells rest on the named close-commit messages and review artifacts; the two cells where the committed records conflict with each other are disclosed in the table's own caption rather than reconciled here.

Table 7. Provenance of project-process figures in §6 (evaluation and results): three of seven claim groups, chosen to span the frozen-protocol, fault-injection, and Round-1-portability claim classes. All seven groups (23 individual claims total) report MATCH; the full seven-row table is reproduced in the supplementary package as Table S15. MV = the Evaluation and Results section (§6). "Live" = re-derived from the working tree this pass.

Claim group in prose	Location	Live-record source	Status
Frozen evaluation snapshot, live-corpus size, and predicate fixtures (3 claims)	MV:14	the frozen protocol record; the checksum manifest; a live re-derivation against both ledgers and the manifest	MATCH
Seeded fault injection and the protocol v1.2 mutation-testing extension (4 claims)	MV:28, 52, 77	the evaluation gate report §7 (223/223 mutants); §9's mutation-testing detail (Tables S8–S10, moved per USER-113)	MATCH
Round 1 portability/sensitivity/agreement checks: OCR-subset, OQ-023 sweep, second corpus, extraction variants, re-implementation, PROV-O/N (6 claims)	MV:181, 189, 197, 206, 214, 230	§9's Round 1 evaluation-detail table (Tables S1–S7, moved per USER-113)	MATCH

Table 8. Provenance of project-process figures in §5 (pipeline architecture) and §7 (bounded self-application case): three of four claim groups. All four groups (12 individual claims total) report MATCH; the full four-row table is reproduced in the supplementary package as Table S16. DM = the Bounded Self-Application Case section (§7); AP = the Supplementary Process Tables appendix (Appendix B).

Claim group in prose	Location	Live-record source	Status
Pipeline governance: director/agents structure, seven named gates, reopened Gate-D's twelve conditions (3 claims)	MP:4; MP:33–35; AP:378–380	the project charter; Table 3; the Gate-D sign-off record	MATCH
Intake pipeline and governance-record counts (2 claims)	MP:86–88; MP:98–104	the intake-skill record; the decision log and task ledger, re-derived 2026-08-24 via `tools/counters.py` (258/109/391)	MATCH
Bounded self-application case: six credibility dimensions, five pipeline stages, cycle-close table, coverage-trajectory replay, sweep growth (5 claims)	DM:17–18, 27–32, 59–62, 62–63; AP:342–344	the capability-verification record; the coverage-trajectory record; the review-trajectory record; four cycle-close commit records	MATCH

A.1 Round 1 Evaluation Detail

§6's Round 1 subsection summarizes seven checks run under Protocol v1.1 (frozen 2026-08-22/23, sha256 `7a32741c...ef2fc8`, a disclosed same-generation addendum that changes no v1 predicate, threshold, or hypothesis). Per author ruling USER-113, the seven per-check tables that used to sit here are reproduced verbatim, from each check's own run record, as Tables S1–S7 in the supplementary package. Every headline figure is still stated in §6's own body prose, and the pointer tables below keep this appendix's own numbering intact for §6's cross-references. Per author ruling USER-135, the table mapping the full evaluation portfolio – fresh-protocol checks, the seven Round 1 checks below, and this round's own live re-derivation and provenance-emission checks alike – to what each strand grounds and its own disclosed limit is reproduced verbatim as Table S17 in the supplementary package (Table 9), before the per-check pointer tables give each one's own detail.

Table 9. Pointers: four evaluation-detail tables moved to the supplementary package (USER-113, USER-135).

Check	Supplementary table
Thirteen-strand evaluation-portfolio map (USER-135)	Table S17
R1-A: OCR-subset re-verification (USER-113)	Table S1
R1-B: full-history census (USER-113)	Table S2
R1-C: OQ-023 threshold-sensitivity sweep (USER-113)	Table S3

OQ-039 live re-derivation. The frozen 771-row figure above predates later ledger growth. A fresh, full recheck as of 2026-08-24 (closing OQ-039, USER-128 option (c)) found 53 quote rows (of 792) and 0 claim rows (of 83 stamped) whose live status disagreed with the stored one. Row-by-row adjudication traced every disagreement to a checker witness-selection gap or a ledger data-entry gap, never a wrong citation (Figure 7

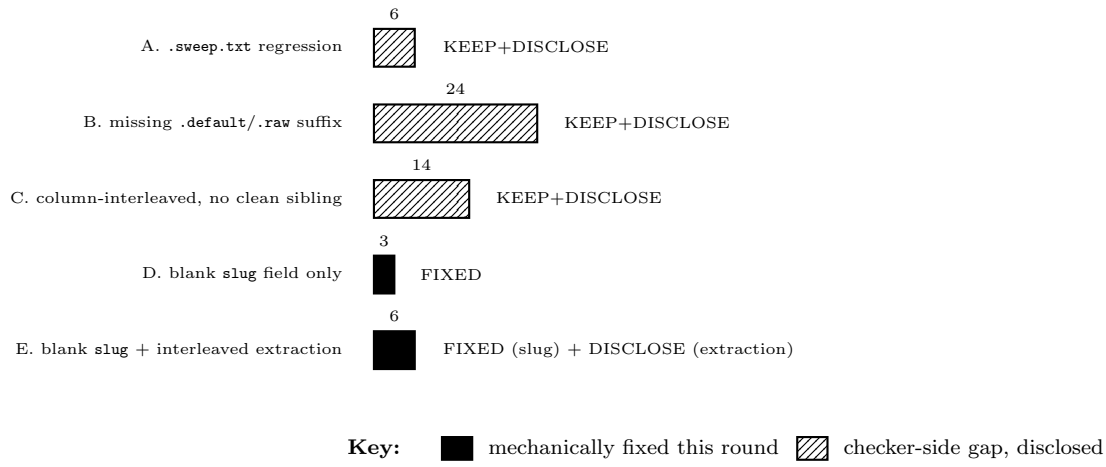

Fig. 7. OQ-039 quote-side disagreements, all 53 rows, by adjudicated cause. A fresh recheck of every `quote_ledger.csv` row against a live re-run of the same predicate found 53 rows (of 791 comparable) where the stored status disagreed with the fresh verdict; row-by-row adjudication traced every one to a cause, never to a miscited source. Bar length is proportional to the row count in each class. Solid bars (classes D and E, 9 rows) are a ledger data-entry gap – a blank `slug` field – mechanically fixed this round, PROV-logged. Hatched bars (classes A, B, C, and E's extraction component, 44 rows) are a gap in the checking machinery itself – the wrong, or a worse, extraction-variant file outranked the one the row's own `verification_method` column names as its original witness – disclosed as a follow-on for `predicates.py`, not fixed here. Class B alone, a missing `.default.txt/.raw.txt` suffix in the extraction preference order, accounts for 24 of the 44 checker-side rows, four times any other single cause. The claim-ledger side of the same check (83 stamped rows) found zero disagreements, not charted here since a single zero has no distribution to show.

below breaks out the five-way cause split); the 9 data-entry cases were corrected, the checker gaps disclosed as a future fix.

Table 10. Pointers: four further Round 1 evaluation-detail tables moved to the supplementary package (USER-113).

Check	Supplementary table
R1-D: second-corpus (MLA) robustness (USER-113)	Table S4
R1-E: extraction-variant sensitivity census (USER-113)	Table S5
R1-F: independent re-implementation agreement (USER-113)	Table S6
R1-G: PROV-O/PROV-N structural validation (USER-113)	Table S7

A.2 PROV Go-Forward: Live Emission

Protocol v1.4 promotes the one-time Round 1 PROV serialization above to a live emission layered onto the invalidation engine's event log. Two firings have run to date: the initial eighteen-row field-integrity repair, and this round's nine-row OQ-039 ledger-metadata repair (Table A.1); together they wrote 27 `wasRevisionOf`-classified events, 128 triples, and an independent re-parse newly failed two of five structural checks – derivation acyclicity and the no-`wasRevisionOf`-cycle check, both on one self-loop, traced to the nine new events' empty `content_hash_before`/`content_hash_after` fields collapsing each entity's before/after URN onto one node. The root cause – an event-log writer hardcoding both hash fields to empty regardless

of what the detector actually knew – was diagnosed and fixed the same round: the writer now records the detector's real hashes (refusing rather than emitting an unrecoverable empty one), and 9 corrective events were appended, never rewriting or deleting the original 9. A regression test (5 new cases) now guards the fixed path. Re-parsed after the fix: 36 events, 182 triples, five of five structural checks pass – unique identifiers (792/792 quotations, 730/730 claims), no dangling references, acyclic derivation, no `wasRevisionOf` cycle (27 edges, verified cycle-free), and the vacuous no-premature-invalidation check (9 `Invalidation` instances, presence-only). The underlying ledger fix itself was never affected (its real hashes are recorded separately, Table A.1); the defect and its repair both sit in the invalidation engine's own event-emission path. The sixth check – generation-before-use timing (Protocol v1.4 §4.1–§4.2) – remains `NOT_RUN`, unchanged.

Disclosed gaps. Protocol v1.4 cites "36 live triggers" already logged as a ready backfill source; no file on disk substantiated that figure when the first emission ran, and this round adds a second, newly-found gap: the structural-check regression above. Both are routed as future work rather than smoothed over.

A.3 Mutation-Testing Extension Detail (Protocol v1.2)

The following figures come from Protocol v1.2's mutation-testing extension (frozen 2026-08-23, sha256 `e3da8869...74dc`) and its independent re-audit. Per author ruling USER-113, the three detail tables that used to sit here ($n = 100$/class scale-up, the 80-item equivalent-mutant audit, the metamorphic-invariance check) live in the supplementary package as Tables S8–S10, verbatim from each run's own record; §6's own body prose states every headline figure directly.

B Supplementary Process Tables

B.1 Cross-Cycle Corpus and Manuscript Growth

The sibling manuscript's growth across its own committed cycles – referenced in §7 as evidence the demonstrated cycle ran at real scale – is reported here as a one-line pointer; per author ruling USER-113 (2026-08-23, task R7.W-APPENDIX), the full per-cycle table (source manifest, quotation-ledger, page-count, and terminal-review figures at each of the sibling manuscript's four committed cycle closes, including two disclosed, unreconciled record conflicts) now lives in the anonymized supplementary package as Table S11 and its CSV twin.

B.2 Repair-Episode Ablation

§6 narrates one of the four dated repair episodes to the mechanical quote-verification sweep in full (episode 1, the column-interleaved-extraction case); per author ruling USER-113, the table giving all four episodes (including the three referenced there only by count) now lives in the supplementary package as Table S12.

B.3 Reviewer-Diversity Corroboration

§5 states the topline Szymkiewicz–Simpson coefficient (0.35) between this paper's external (cross-family) and internal (seven-dimension) AI-review registers, and both readings of that coefficient, in body prose. Per author ruling USER-113, the granular breakdown behind that coefficient now lives in the supplementary package as Table S13.

Table 11. Pointers: three supplementary-package tables referenced above (USER-113).

Table referenced above	Supplementary table
Cross-cycle corpus-and-manuscript growth	Table S11
Four-episode repair-episode ablation	Table S12
Reviewer-diversity granular breakdown	Table S13

B.4 Sibling-Manuscript Gate Episodes

§5 illustrates the pipeline's cyclicality with two episodes from its sibling manuscript, a humanities research paper the same architecture also produced, not this paper itself, and gives both in full here.

Gate D, thesis selection, was not a one-time hurdle there: when that paper's central argument changed materially after Gate D had already passed once, the pipeline did not carry the earlier ratification forward. It reopened Gate D, required a second, independently commissioned sign-off under the new framing, and recorded a new pass state — twelve conditions, four blocking — before later work could depend on the revised argument.

Gate F, the terminal adversarial review, shows the same discipline in reverse. Across two successive cycles it returned major revision. Most recently it returned major revision across five argument dimensions with sixty-one findings, eleven severe. It was recorded as convened but *not* passed. Sign-off was withheld pending the director's own read-through. The director then confirmed this ruling explicitly, not by default. The surviving record of that episode is itself a reconstruction, assembled after the fact from sub-agent logs once the contemporaneous review report turned out never to have been written, disclosed here because surfacing gaps in its own record is the architecture's own standing claim.

B.5 Review-Round Findings Trajectory

§5 points here for the full per-round breakdown of this paper's own Cycle-8 review chain (author ruling USER-110, task R6.W-ARCH-FIG, 2026-08-23: Figure 2 and Figure 3 stay in the body; this figure alone moves here). The caption, alt-text description, and every plotted number are unchanged from the body-figure version this replaces.

B.6 Self-Application as a Live Evidentiary Record

§7 states the general limit of a self-applied case: it licenses a claim about mechanism, never a claim about generalization. Per author ruling USER-137 (2026-08-24, superseding the per-round appendix-length fights under USER-120/USER-135), this section carries a FIXED-SIZE summary rather than a growing per-step record: one row per event class, with each row's count updated in place as further Cycle-8 steps occur, never by adding a new row. The unabridged, individually dated record – 36 steps as of this round, spanning Round 1's frozen-protocol governance through this round's own appendix restructuring – is Table S14 in the supplementary package (`self_application_full.csv`), which grows every round; this table does not. The classification below follows an autobiographical-design/research-through-design literature review's convergence on exactly this kind of auditable compression [Desjardins and Ball 2018; Flyvbjerg 2006; Krebs 2026; Neustaedter and Sengers 2012; Zimmerman et al. 2007] – also called dogfooding [Silva et al. 2019] or an annotated portfolio [Gaver and Bowers 2012].

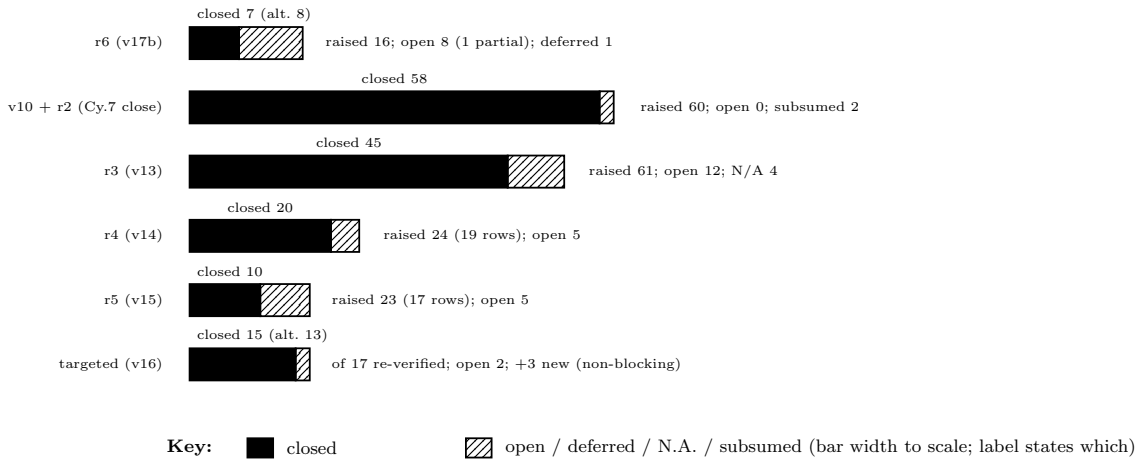

Fig. 8. Findings raised, closed, and open across this manuscript's review chain to date, from Cycle 7's joint v10-adversarial-plus-AI-review-r2 disposition (closed 2026-08-20) through Cycle 8's ongoing review-and-remediation chain on a single build, current through Round 4's close; at least five further review rounds since then are not yet reflected below and need a fuller rebuild of this figure, not only a caption update. Bar length is proportional to the raised count within each row; the solid segment is closed, the hatched segment is everything not closed (open, deferred, not-applicable, or subsumed, per that round's own classification — the exact split is given in the adjacent label, not encoded by hatch sub-type). Severity taxonomies differ across rounds and are not forced into one scale: the v10+r2 and v13 rounds classify findings as submission-blocking, mechanical-fix, substantive, author-governed, rejected, subsumed, or deferred, with no S0–S3 label; the v14, v15, v16, and r6 rounds use S0–S3. Three rows carry a disclosed denominator or count mismatch rather than a forced reconciliation. r4's closure report tallies 29 outcomes against a pre-deduplication count of 24 raw claims rather than the 19 post-deduplication governed rows the same round's disposition table names. The targeted v16 pass's own closed count is reported as 15 by its dedicated closure report but as 13 by a later same-day synthesis report re-deriving the identical 17 rows. r6's own closed count is likewise reported as 7 by a dedicated closure report's row-by-row severity table but as 8 by a later same-day round-report synthesis; in every one of the three cases, both figures are given rather than one silently discarded. Open S0/S1 counts, where that scheme applies, run as follows. v14: 0 open S0, 2 open S1 (DT-02, DT-16). v15 at its own close: 1 open S0 (R2D-01), 1 open S1 (R2D-02). Targeted v16: 0 open S0, 1 open S1 (R2D-02, unowned across three consecutive rounds). r6/v17b: 2 open S0 (AS6-01, AS6-02, both a package-integrity defect the review independently re-confirmed rather than took on trust) and 5 open S1 (four numbered rows plus one author-ruled row whose routing memo was never drafted). The v10+r2 and v13 rounds report a submission-blocking count instead: 2 of 2 closed, and 3 of 3 closed, respectively. The targeted v16 pass re-verifies v15's 17 rows rather than raising a fresh full set; it separately raised 3 new, non-blocking findings (two minor, one moderate) of its own. The r6/v17b row's own manuscript PDF is independently reported clean across every mechanical axis this same round (0 build errors, 0 overfull boxes, citation parity, a clean locator audit, and a clean identifier sweep of the rendered text). Every one of its open S0/S1 findings concerns the separately-distributed artifact package or the governance records instead, not the manuscript PDF itself.

What this table cannot show. A capped, class-level count trades away exactly what the uncapped record showed: which specific step produced which specific figure, and how one step's limit qualifies another's claim. That detail is not lost, only relocated – every count above is reproducible against Table S14 and the live project files named in each row's own evidence pointer, and Table S14 itself keeps growing every round under the same schema this table replaces. Two counting choices are disclosed rather than hidden: several classes use a mechanical proxy over the project's governance log (a keyword match against dated rows) rather than a hand-classified census of every individual event, so a count states what that proxy measures, not a claim that no borderline row was mis-bucketed; and this table's own governing rulings, USER-137 (this rule) and

Table 12. The self-application case, capped at one summary row per event class (USER-137). Counts are re-derived live against the project's own records each time this table is rebuilt, not carried forward from a prior report; the method behind each count is named in the Evidence column rather than asserted bare. Every class's latest event falls on this round's own close date because several classes are touched by the same omnibus Round-13 handoff row in the project's governance log; this is a property of that log's own recent density, not a rounding convention.

Event class	Count	Latest	Evidence pointer
Governance rulings	116	2026-08-24	DECISIONS.md, dated `**USER-NN**` rows (114 distinct ids); includes USER-137 and USER-138, this round's own appendix and word-ceiling rulings
Records repairs	35	2026-08-24	DECISIONS.md, dated rows stating a correction to a prior figure or claim
Anonymity events	19	2026-08-24	DECISIONS.md, dated rows addressing an anonymity-scrub or anonymity-risk finding
Tooling adoptions	8	2026-08-24	Table S14 rows 21, 25, 28, 32: the capability-gap harvest (six builds), the invalidation-engine promotion, the method-template extraction
Evaluation runs	14	2026-08-24	named study directories under the evaluation results tree, Rounds 1–12b
Review rounds	14	2026-08-24	fourteen dated review-round directories, drafts v13–v26
External intakes	5	2026-08-24	source-manifest growth, 174 → 193 → 208 → 215 → 225 → 245, five dated batch reports
Incident-catches	9	2026-08-24	DECISIONS.md, dated rows naming a defect caught by the project's own gate or audit
Publications	1	2026-08-24	DECISIONS.md USER-134: the public method-template repository (published, then renamed)
Adjudications	11	2026-08-24	eleven dated disposition-table passes, drafts v13–v25
Schema migrations	13	2026-08-24	DECISIONS.md, dated rows naming a ledger, manifest, or protocol schema change
Handoffs	24	2026-08-24	DECISIONS.md, dated rows closing one round and opening the next

USER-138 (a durable word-ceiling re-ratification), are logged as their own dated DECISIONS.md rows by a sibling records task the same round and are counted in the governance-rulings row above, not excluded from it.

B.7 Cell-Level Provenance Census

§8 states the headline result of a full-history, cell-level provenance census across both ledgers and the source manifest, moved here per this round's own placement audit (corroborating detail, not an argument-bearing claim). Replaying every git-committed state end to end, not sampling a subset, the census found 94.4% of 5,520 historical cell changes traced to a dated correction record under a broad match, and 58.8% under a strict exact-cell match (same row, column, and commit). The shortfall concentrates in one commit predating the project's own records tooling; every later commit clears 99.6% or better under the strict count. This extends the self-test suite's ninth test (a silent-edit census at draft scale) to the ledgers' full committed history; the figures are unchanged from where they sat in the body before this move.