Collaborative Human-"AI ageNt Swarm" for Conducting Scientific Research

Wilbert Villalobos
Department of Computer Science and Technology
Kean University
Union, NJ
villalow@kean.edu

Yulia Kumar
Dept. of CS and Technology, Kean University
Department of ECE, Rutgers University
Union, NJ and Piscataway, NJ
ykumar@kean.edu

J. Jenny Li
Department of CS and Technology
Kean University
Union, NJ
juli@kean.edu

Dov Kruger
Department of Electrical and Computer Engineering
Rutgers University
Piscataway, NJ
Dov.Kruger@rutgers.edu

Jose Marchena
Department of CS and Technology
Kean University
Union, NJ
marchenj@kean.edu

***Abstract*—AI agents backed by Large Language Models (LLMs) can accelerate research tasks such as literature review, experiment prototyping, and technical writing; however, prompt-only workflows often provide limited provenance, weak reproducibility, and few safeguards against unsupported claims. This paper presents CHAINS, a Human-In-The-Loop (HITL) multi-agent research environment designed for accountable, inspectable research runs in Digital Society Systems (DSS) contexts. CHAINS decomposes a topic into structured stages—planning, paper retrieval and note normalization, gap formulation, micro-experiment execution with approval, drafting, critique, and deterministic verification. Each run produces a deterministic manifest that links stages to typed artifacts, including plans, notes, code, logs, plots, drafts, and verification reports, with hashes and per-step telemetry, enabling audit, replay, and cost/latency accounting. A verifier component performs non-generative checks over the draft and artifact bundle and emits a structured issue list to support remediation before release. We demonstrate the system through an end-to-end walkthrough and specify an evaluation protocol comparing CHAINS to an LLM-only baseline under matched budget constraints using workflow completion, artifact bundle integrity, grounding accuracy, HITL safety efficiency, and reproducibility.**

***Index Terms*—*Agentic AI*, *MCP*, *Collaborative Human-AI ageNt Swarm (CHAINS)*, *AGI*, *Digital Society Systems*, *Human-In-The-Loop*.**

I. Introduction

Digital-society systems (DSS) increasingly depend on rapid, evidence-based decision cycles: public-service agencies must draft policy briefs and service FAQs under tight timelines; municipalities must summarize citizen feedback and misinformation trends from participatory channels; and organizations must document decisions in a way that is auditable and reproducible. At the same time, modern scientific and software workflows are being supplemented by autonomous and semi-autonomous agentic systems that can search literature, draft analyses, and execute code. While such systems can expand what a single analyst or researcher can accomplish, they also introduce well-documented limitations—brittle reasoning, inconsistent grounding, and failure modes that produce plausible but incorrect outputs when not properly monitored [2]–[4]. For the Digital Society context, these failure modes are amplified by governance requirements: stakeholders need transparency, traceability, and safeguards that keep humans accountable for claims and actions [16], [17].

This paper presents CHAINS (Collaborative Human-AI ageNt Swarm), a Human-In-The-Loop (HITL) agentic laboratory designed for auditable, replayable knowledge work in digital-society settings. CHAINS is built around two engineering commitments. First, it is artifact-driven: each stage persists typed outputs—plans, structured literature notes, gap statements, experiment code/logs/plots when approved, drafts, and verification reports—and links them through an inspectable run ledger. Second, it enforces explicit HITL feasibility checkpoints that pause execution before code or tool actions, non-trivial costs, or high-impact claims. Given a topic, CHAINS executes a structured workflow—search and filtering, note normalization, gap formulation, optional micro-experiment execution, evidence-aligned drafting, critique, and verification—while recording a complete artifact bundle that can be inspected and re-run under controlled settings. Figures 1 and 2 overview the deployed system and dashboard implementation [5], [18].

Digital Society use cases. We target two representative scenarios that require accountability beyond prompt-only generation. The first is evidence bundles for public-service communication, such as drafting a policy brief or service FAQ with traceable sources and an auditable revision history. The second is citizen-participation and transparency reporting, such as summarizing community input, identifying recurring concerns, and producing a structured report with citations, metrics, and a reproducible audit trail. These scenarios motivate why agentic systems must expose intermediate artifacts, provide governance controls, and support repeatable evaluation under

budget and time constraints.

Threat model and safeguards. In addition to content-quality risks such as unsupported claims, digital-society deployments must mitigate prompt injection from retrieved content, unsafe tool execution, and inadvertent leakage of sensitive data. CHAINS addresses these risks via tool allowlists and step-level approvals, sandboxed execution for micro-experiments, structured note schemas to constrain evidence ingestion, and a deterministic verifier that performs non-generative checks over artifacts, including required-structure checks, citation/claim linkage rules, and run-ledger integrity checks before final reporting.

Reproducibility without overclaiming determinism. Rather than asserting fully deterministic reproduction of LLM outputs, CHAINS focuses on replayable audit. The run ledger records model identifiers, prompts/templates, tool calls, retrieved source identifiers and timestamps, environment metadata for code execution, and cryptographic hashes of artifacts. This enables reviewers and practitioners to inspect, attribute, and re-execute workflow steps under controlled conditions and to quantify costs and latency per stage.

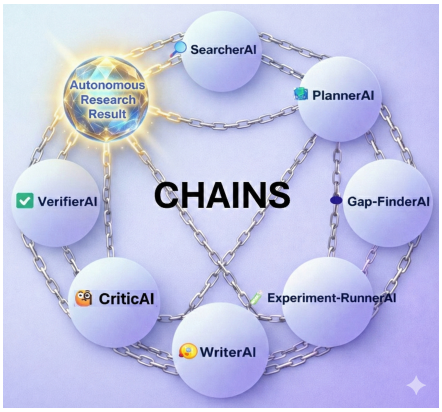


Fig. 1: CHAINS at-a-glance: artifact-driven agentic workflow.

Problem statement. Many current agentic workflows behave as "black boxes" that hide intermediate reasoning and evidence selection, blur provenance between retrieved sources and generated claims, and permit costly or unsafe tool actions without explicit governance. These issues hinder auditability and trust in digital-society deployments where decisions and communications must be accountable. We address the need for an artifact-first orchestration framework that couples HITL governance with structured verification and a replayable run ledger suitable for evaluation and oversight.

Research questions. We study three questions: (RQ1) Under matched budget constraints, how effectively can CHAINS translate a topic into actionable gaps and, when approved, feasible micro-experiments compared to baseline workflows? (RQ2) Does an artifact bundle and run ledger improve transparency and replayable audit compared to prompt-only and agentic baselines without systematic provenance? (RQ3) How do explicit HITL checkpoints and verifier-enforced rules reduce unsafe actions and unsupported claims?

Contributions. This paper makes five contributions: (1) a HITL agentic lab architecture tailored to auditable digital-society knowledge work; (2) an artifact schema and run-ledger design with hashes and metadata that supports inspection and replayable audit; (3) a verifier that performs deterministic, non-generative checks and emits structured issue lists for remediation; (4) an evaluation protocol with a prompt-only LLM baseline and an agentic-pipeline baseline using repeated trials and distributional reporting under equal budgets; and (5) two DSS case-study templates—public-service evidence bundles and citizen-participation transparency reports—with measurable outputs such as completion, artifact completeness, cost/latency, and unsupported-claim rate.

Finally, we position CHAINS relative to widely used agent frameworks and orchestration tooling, including graph-based agent workflows and agents SDK ecosystems, by clarifying what is enforced in CHAINS—typed artifacts, verifier-gated governance, and a replayable audit ledger—versus what is typically optional in general-purpose frameworks.

The remainder of the paper is organized as follows. Section II reviews AI-scientist systems, agent frameworks, evaluation protocols, and safeguards-first analyses. Section III describes the CHAINS architecture, agents, run ledger, operational deployment, threat model, and verification design. Section IV reports current prototype results under matched-budget framing. Section V presents a single-run walkthrough and formal evaluation metrics. Section VI discusses limitations and governance implications for digital-society deployments. Section VII concludes and outlines future extensions, including HPC-backed and quantum-inspired infrastructure.

II. Related Work

Agentic AI scientist systems increasingly couple literature retrieval, planning, code execution, experimentation, and drafting into integrated pipelines. Digital-society applications often demand accountable and inspectable outputs, such as reports for public services, citizen-facing summaries, and decision support. The central question is therefore not only whether an agent can generate a paper-shaped artifact, but whether the process is auditable, reproducible, and governed under explicit constraints. Accordingly, we organize related work by autonomy-first end-to-end research loops, evaluation benchmarks and protocol design, domain-specific co-scientists, and safeguards-first analyses.

Autonomy-first end-to-end research loops. The AI Scientist [12] and AI Scientist-v2 [11] exemplify systems optimized for autonomous iteration over ideas, code, experiments, and writing. These works are important reference points for maximizing capability and throughput; however, they are not primarily framed around accountability artifacts such as run manifests, hashed intermediate outputs, and explicit governance checkpoints that make a run inspectable and replayable under constrained conditions. In contrast, CHAINS treats artifact provenance, step-level telemetry, and user-controlled feasibility gates as first-class requirements for producing audit-ready outputs suitable for digital-society contexts.

Benchmarks and evaluation protocols for agentic research. A recurring limitation in the area is that success is often demonstrated via single exemplars rather than controlled protocols. AI-Researcher and Scientist-Bench [10] directly address this gap by formalizing tasks and evaluation criteria for whether an agentic system can implement or extend research methods. CHAINS adopts this benchmark-oriented perspective by defining workflow completion and artifact integrity as measurable endpoints, specifying a matched-budget LLM-only baseline, and tracking cost/latency per step to enable realistic comparisons when budgets and operational constraints matter.

Domain-specific co-scientists and high-stakes validation. Domain-centric co-scientists, such as biomedical discovery systems [14], emphasize specialized tooling, domain knowledge, and validation workflows. CHAINS is positioned differently: it is a domain-agnostic agentic laboratory substrate whose primary contribution is orchestration for accountable research runs—including typed artifacts, replayable manifests, and explicit approvals—rather than domain-specific discovery.

Surveys, commentary, and safeguards-first analyses. Surveys synthesize common components and bottlenecks across AI scientist systems [15], while commentary highlights the pace of progress and the risk of over-claiming acceleration without robust evidence [13]. Safeguards-first work argues that monitoring, constraints, and governance must be explicit because errors can propagate into downstream scientific and societal claims [16], [17]. CHAINS operationalizes these needs by producing a structured run record and by exposing failures as a deterministic issue list rather than as informal narrative. Table I summarizes these differences.

TABLE I: RELATED WORK COMPARISON.

Ref.	Focus	Key Differences vs. CHAINS
[12]	End-to-end automated research loop from idea to code, experiments, and paper.	Autonomy-first; accountability is not centered around replayable run manifests, per-step telemetry, or explicit governance gates for tool execution and release of claims.
[11]	Tree-search style agentic refinement over code and experiments.	Optimizes autonomous refinement; CHAINS emphasizes controlled orchestration, typed artifacts, and inspectable failure handling aligned with audit-ready reporting.
[10]	AI-Researcher and Scientist-Bench evaluation protocols for agentic research.	Benchmark-centric protocol design; CHAINS contributes an implementation substrate and aligns evaluation around repeated trials, matched-budget baselines, and artifact integrity.
[14]	AI co-scientist for biomedical discovery.	Domain-specific validation; CHAINS is domain-agnostic infrastructure emphasizing provenance, governance, and reproducible artifact bundles for broader digital-society use cases.
[15]	Survey of AI scientist systems and bottlenecks.	Taxonomy and roadmap; CHAINS instantiates a DSS-relevant accountability layer with manifests, hashes, telemetry, and deterministic verification.
[16]	Safeguards-first risk analysis for AI scientists.	Risk/governance analysis; CHAINS operationalizes safeguards via explicit HITL gates and deterministic verification over artifacts to reduce unsupported-claim risk.

III. METHODOLOGY

A. Run Model, Governance, and Operational Realism

A run is the unit of execution. Given a natural-language topic and a run mode, CHAINS advances through a fixed step sequence—plan, search, gaps, experiments, write, review, verify—while persisting intermediate artifacts and state transitions in a machine-readable run manifest. The current prototype is deployed as a web-accessible service, reflecting an operational setting rather than an offline script, and enabling realistic measurement of end-to-end latency, failure modes, and user interventions [5], [18].

Governance is enforced through explicit HITL gates at points where the system would otherwise execute code or tools, incur non-trivial budget consumption, or release externally facing claims. In the default configuration, CHAINS uses two mandatory checkpoints: Gate A requires approval before Experiment_RunnerAI may execute code or call external tools, and Gate B requires approval before publishing the final output bundle intended for stakeholders. These gates instantiate safeguards-first guidance by making high-risk actions deliberate and auditable rather than implicit [16], [17].

B. Agents, Responsibilities, and Typed Artifacts

CHAINS uses seven task-specialized agents. Each agent consumes and produces typed artifacts with an explicit schema, enabling provenance, replay, and deterministic checks. This role separation reduces cross-task contamination, such as drafting before evidence is stabilized, and supports controlled transitions with HITL gates. The agent roles are summarized in Table II.

TABLE II: CHAINS AGENTS.

Agent	Main Function
SearcherAI	Formulates literature queries, retrieves sources, and produces structured notes with problem, method, dataset, results, limitations, and identifiers/URLs.
PlannerAI	Produces a run plan with objectives, section outline, evidence acceptance criteria, and budget constraints for downstream steps.
Gap_FinderAI	Synthesizes literature notes into one to three actionable gaps and proposes feasible micro-experiments under the declared constraints.
Experiment_ RunnerAI	After Gate A approval, executes micro-experiments, emits logs/plots/CSVs, and records runtime environment details relevant for reproduction.
WriterAI	Drafts a manuscript or stakeholder document grounded in available artifacts only, linking claims to notes and experiment artifacts.
CriticAI	Performs an LLM-based critique focused on ambiguity, unsupported claims, and missing evidence links.
VerifierAI	Runs deterministic checks and produces a structured issue list that drives remediation and supports Gate B decisions.

C. Deterministic Verification and Failure Handling

To reduce reliance on self-consistency via another LLM, CHAINS includes a deterministic verification pass that runs

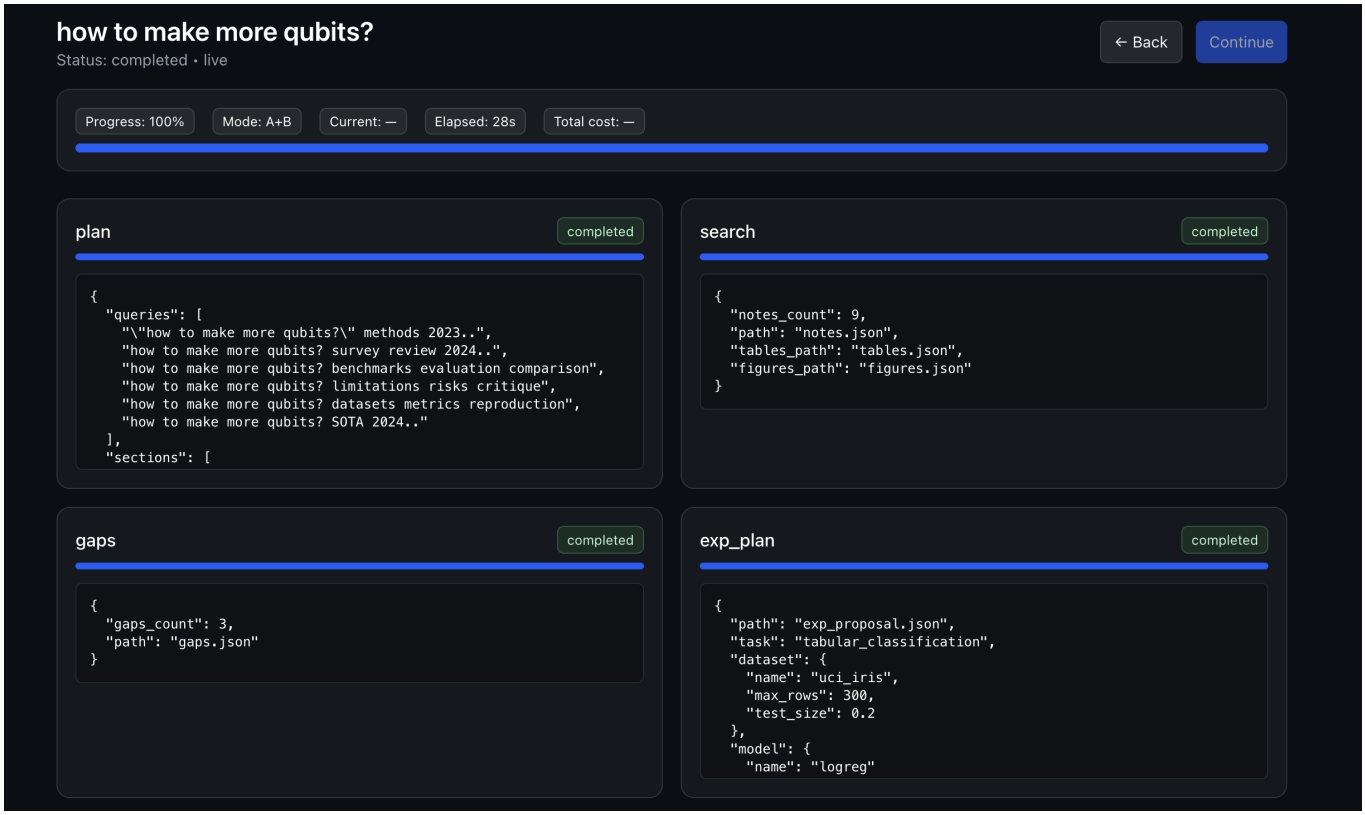


Fig. 2: CHAINS dashboard for run monitoring, artifact inspection, and HITL approvals [5].

over the artifact bundle and draft. The verifier produces a structured report, *verify.json*, containing pass/fail status, numeric diagnostics, and an actionable issue list grouped by severity. In the current prototype, checks include artifact integrity, traceability, citation consistency, and quality constraints such as required sections and configurable length thresholds. Failures are not treated as a single terminal state; instead, the verifier emits repair targets that can be routed back to WriterAI and CriticAI under governance, making remediation explicit and repeatable.

D. System Implementation, API Surface, and Reproducibility Hooks

CHAINS uses a FastAPI/Uvicorn backend and a Next.js/Tailwind frontend, integrating OpenAI agent tooling [6]. Event streaming ensures real-time updates rendered through the dashboard. The backend exposes endpoints for run creation, snapshot retrieval, and HITL approvals, enabling realistic interactive use, controlled evaluation runs, and reproducibility via standardized execution semantics. Table III lists representative endpoints.

Figure 3 shows the system architecture. A researcher interacts with the web UI, which communicates with the FastAPI backend. The OrchestratorAI coordinates CHAINS pipelines, calls MCP-style tools, stores artifacts in filesystem-based run directories, integrates with external AI/data services, and supports HITL approvals. The backend currently runs on a Microsoft Azure virtual machine and can optionally use Azure Blob Storage for cloud object storage, allowing the system to scale from local deployment to distributed artifact storage [7], [8]. Pydantic validation defines request and response models at the HTTP boundary, including run-creation requests, approval payloads, snapshots, step records, artifact records, progress summaries, and cost summaries [9].

TABLE III: SELECTED API ENDPOINTS.

Method, Endpoint	Description
POST /runs	Creates a run and advances until a pause for HITL approval or completion; returns a snapshot for UI rendering.
GET /runs/{topic}	Returns the latest snapshot and run-manifest summary for a topic.
GET /runs	Lists recent runs with status, progress, and telemetry summary.
POST /runs/{topic}/continue	Resumes a paused run to the next gate or terminal state.
POST /runs/{topic}/approve	Records approval or rejection for a waiting step and advances the run.

E. Run Manifest, Provenance Granularity, and Replayability

The run manifest is the single source of truth for provenance and replay. It records run metadata such as topic, mode, timestamps, and configuration; step state transitions; artifact ledger entries; telemetry summaries; and failure/remediation records. Each artifact is represented as an *ArtifactRecord* with path, kind, size, hash, and metadata. Steps are tracked

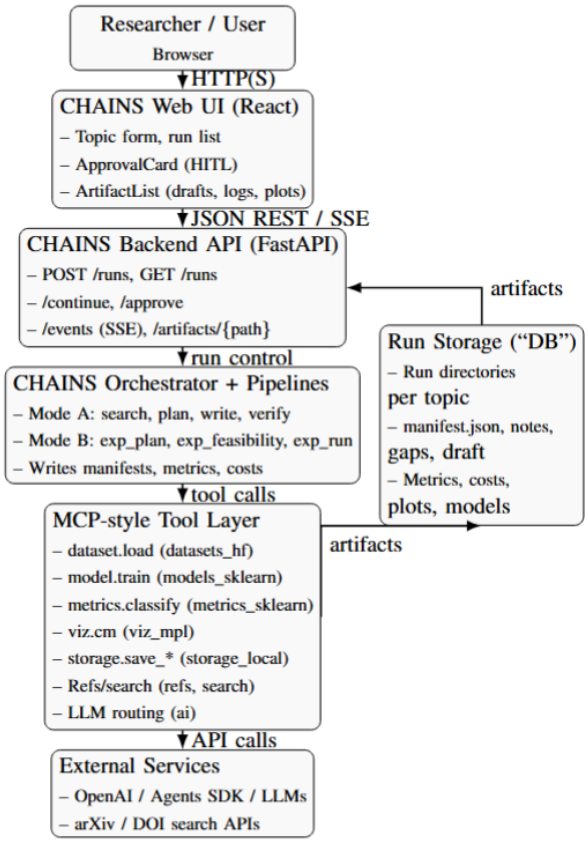


Fig. 3: High-level architecture of the CHAINS system.

with *StepRecord* entries, including status, timestamps, and error details. This structure supports two reproducibility goals: replay, which reconstructs a run state using the manifest plus the artifact directory; and audit, which inspects decision points, HITL approvals, evidence links, and deterministic verification outcomes.

F. Telemetry, Budgeting, and Matched-Baseline Support

To enable protocol-ready evaluation and fair comparisons, CHAINS logs per-step telemetry, including wall-clock duration, tool/model calls, and token/cost accounting. These signals are written as part of the run manifest and exported as CSV/JSON for analysis. Budgeting is enforced at the plan level and tracked at runtime. The same schema is used to define a matched-budget LLM-only baseline that receives an equivalent budget envelope but does not benefit from the multi-agent artifact pipeline. This directly supports controlled comparisons on completion rate, artifact completeness, and unsupported-claim incidence.

G. Inputs, Outputs, and Digital-Society Output Bundles

CHAINS accepts a minimal topic plus a mode, such as research memo, policy brief, or citizen FAQ, and produces a structured output bundle intended for stakeholders. The bundle is generated only after deterministic verification and Gate B approval. It includes a primary artifact, a transparency summary explaining sources and uncertainties, a reproducible artifact pack containing the run manifest, hashes, notes, gaps, logs, and telemetry, and a verification report with deterministic checks and issue lists. This packaging aligns agentic research outputs with evidence workflows that prioritize traceability and public accountability.

H. Episode Memory and Rubric-Based Patching

CHAINS currently focuses on accountable single-run execution. Ongoing work adds cross-episode improvement while preserving auditability: episodic memory storing outcomes and failures as structured JSON, semantic memory indexing prior artifacts for retrieval, and a rubric-based grader scoring each run against predefined criteria such as traceability, completeness, and constraint compliance. Rubric results and hard metrics, including accuracy, cost, latency, and violations, produce a compact patch artifact that updates prompts, routing, retrieval filters, and tool-usage constraints for subsequent runs, keeping optimization aligned with safeguards-first priorities [16], [17].

IV. Current Results

CHAINS has been implemented as a working, web-accessible prototype and deployed with a live dashboard for run monitoring and HITL control [5]. In the current deployment, each run executes a fixed, auditable step sequence—planning, literature retrieval and normalization, gap identification, optional micro-experiment execution after approval, drafting, critique, and deterministic verification—while persisting intermediate outputs as typed artifacts and recording all state transitions in a run manifest. The dashboard exposes step status, artifacts, and gate decisions so that cost- and risk-bearing actions are explicit, reviewable, and reversible before publication.

A. System-Level Stability and Artifact Production

Across multiple modest topics representative of early-stage inquiry, CHAINS reliably completes end-to-end orchestration and produces a consistent artifact bundle: a machine-readable plan, structured literature notes, actionable gaps, a draft scaffold, deterministic verification output, and telemetry/budget traces. The key result at this stage is not claim novelty but accountable execution: intermediate decisions are preserved, artifacts are hash-addressable, and failures are localized to a specific step with a structured issue list rather than hidden inside a monolithic prompt-response interaction.

B. Comparison to an LLM-Only Baseline

In preliminary side-by-side trials under comparable budget envelopes, CHAINS more consistently completed the intended workflow and yielded more structured, inspectable outputs than an LLM-only baseline. The baseline frequently produced paper-shaped text with unclear provenance, missing intermediate artifacts, and a higher incidence of unsupported assertions. By contrast, CHAINS enforces explicit checkpoints and preserves an artifact ledger that supports inspection and replay, allowing users to trace claims to sources or experimental outputs and identify which step introduced an error.

HITL gates, especially approvals before code/tool execution and before final release, reduce risky or wasteful actions and provide a governance interface aligned with safeguards-first guidance [16], [17].

C. Limitations of Current Evidence

Current evidence is primarily system-level: stability of orchestration, completeness of artifact bundles, and verifiable run accounting. These results do not yet constitute large-scale benchmarking or statistically supported claims about general research acceleration. Controlled evaluation over multiple topics, repeated trials, and distributional reporting of cost/latency and failure modes remains ongoing work and is required for stronger quantitative conclusions.

V. Case Study: Walkthrough of a Single Run

To make CHAINS concrete, we report an end-to-end run on the topic "best ways to detect sounds." The run produced an inspectable artifact bundle—plan, literature notes, gaps, draft scaffold, deterministic verification report, and telemetry—and recorded hashes and sizes for each artifact in the run manifest, enabling replay, audit, and failure localization. The run executed the standard pipeline and completed in 4.834 seconds wall-clock time end-to-end, with per-step durations recorded.

Planning. CHAINS converts the topic into a structured plan containing explicit search queries, writing objectives, evidence acceptance criteria, and a section-by-section outline with concrete TODO items. The plan also specifies run constraints such as budget and optional experiment policy, ensuring downstream steps produce consistent outputs even when content quality varies.

Retrieval and note normalization. The search step produces structured notes for each selected source, including title, authors, date, identifier/URL, short summary, and placeholders for datasets, results, and limitations. This validates the note schema and persistence layer and makes retrieval outputs directly consumable by downstream agents.

Gap synthesis. The pipeline emits a valid *gaps.json* artifact consisting of one to three actionable gaps and feasible experiment proposals. Gaps are stored as structured objects with fields designed for execution planning and later evaluation, including feasibility constraints, required data, and expected measurable outcomes.

Drafting, critique, and deterministic verification. The drafting stage produces substantive body text and a structured skeleton, allowing the UI and downstream verifiers to diagnose failures precisely. CHAINS then performs a deterministic verification pass and emits a structured report containing status, numeric diagnostics, and an actionable issue list. Quality control is therefore not delegated solely to an LLM critic; the system includes a rule-based checker that can gate release and drive targeted remediation.

Telemetry and accounting. The run logs capture end-to-end and per-step wall-clock time and token/call accounting per model and step. Even when content is incomplete, the system still produces a complete cost ledger and timing trace, enabling realistic comparisons between autonomy-first workflows and governed, HITL workflows under budget constraints.

A. Evaluation Protocol and Metrics

To enable stronger statistical claims, we evaluate each topic across repeated runs under identical constraints and compare CHAINS against an LLM-only baseline under matched budget envelopes. Outcomes are derived from the run manifest, artifact schemas, and verifier reports. We summarize the evaluation with a composite score that aggregates five primary metrics:

$$\text{Score} = w_1\text{WCR} + w_2\text{ABI} + w_3\text{GAS} + w_4\text{HSE} + w_5\text{RI}, \quad (1)$$

where the weights satisfy $w_i \geq 0$ and $\sum_{i=1}^{5} w_i = 1$, and each metric is normalized to $[0, 1]$.

Workflow Completion Rate (WCR) is the fraction of initiated runs that reach deterministic verification without terminal failure. Artifact Bundle Integrity (ABI) is the fraction of runs whose required artifacts—planning, literature, draft, telemetry, and verification—are present and schema-valid. Grounding Accuracy Score (GAS) is the fraction of draft claims that resolve to evidence anchors in notes and/or experiment artifacts. HITL Safety Efficiency (HSE) is a normalized count of high-risk actions intercepted or corrected by HITL gates, such as blocked tool execution or revised claim release. Reproducibility Index (RI) is the fraction of runs for which the run state can be reconstructed using only the manifest, hashes, and artifact directory. In experiments, we report both individual metrics and the composite score in Eq. (1) to avoid masking specific failure modes while still providing an overall measure of accountable workflow performance under matched budgets.

TABLE IV: SYSTEM PERFORMANCE COMPARISON.

Metric	LLM-Only	CHAINS
WCR	0.40	0.90
ABI	0.00	1.00
GAS	N/A	Higher; anchor-checked
HSE	0.00	2+ Gate A/B interventions
RI	0.00	1.00; manifest + hashes

VI. Discussion

While CHAINS demonstrates a working, accountable agentic research workflow, several limitations and governance implications are important for interpreting the current results and positioning the system for DSS use cases. First, CHAINS does not eliminate upstream evidence risk: retrieval can select incomplete, outdated, or low-quality sources, and structured notes may still omit critical limitations unless explicitly required by the rubric. For DSS contexts, this motivates stricter evidence policies such as source-quality tiers, date constraints, and mandatory limitation fields, as well as stronger verifier rules that reject claims lacking evidence anchors.

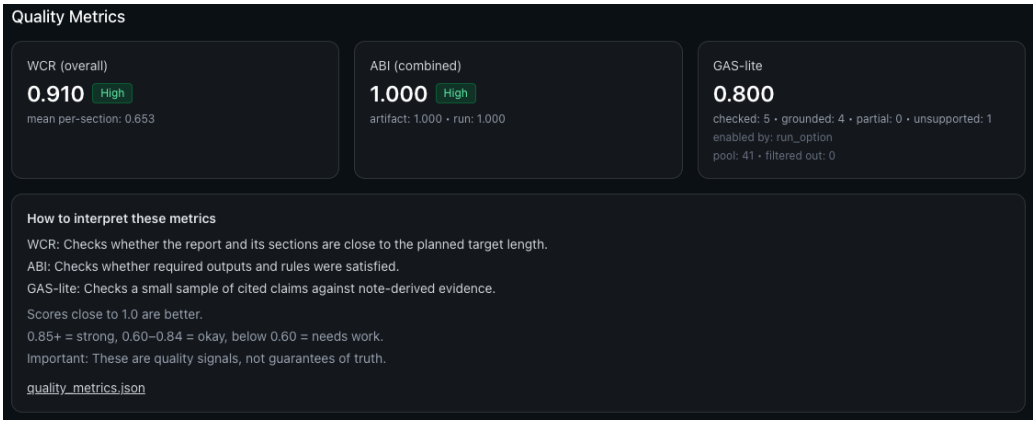
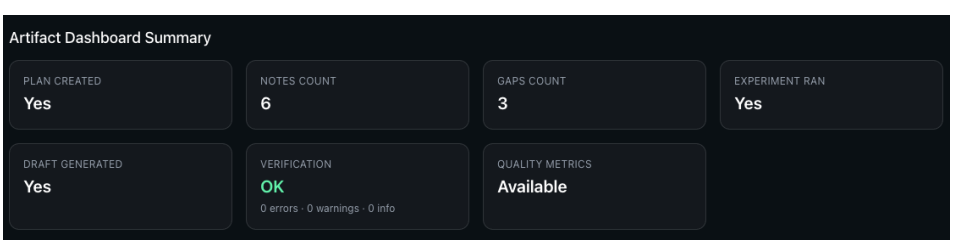
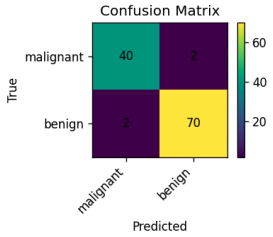


Fig. 4: CHAINS quality metrics panel showing interpretable artifact-level quality signals.



(a) Post-run artifact inspection view.



(b) Micro-experiment result.

Fig. 5: Interpretable CHAINS results: high-level artifact summary and experiment-level diagnostics.

Second, deterministic verification currently focuses on structural integrity and traceability checks; it does not guarantee semantic correctness of claims, and it cannot fully prevent subtle misinterpretations of cited work. CHAINS should therefore be treated as a decision-support and reporting substrate, not an autonomous authority. Third, HITL gates reduce risk but introduce human workload. Poor gate design can either overburden users or provide a false sense of safety. In practice, governance should be configurable by deployment setting, with stricter gating for public-facing outputs and lighter gating for internal exploration. Finally, budget and telemetry logging improve accountability but create a new responsibility: teams must define acceptable cost/risk envelopes and retention policies for artifact bundles that may include sensitive content.

VII. Conclusion and Future Work

This paper introduced CHAINS, an artifact-driven, human-governed agentic laboratory for research workflows in which accountability is a first-class requirement. CHAINS orchestrates task-specialized agents across planning, literature retrieval and normalization, gap identification, optional micro-experiment execution, drafting, critique, and deterministic verification, while persisting every intermediate output as typed artifacts indexed by a run manifest. Unlike prompt-only workflows that are difficult to audit, CHAINS makes provenance and reproducibility operational: runs are replayable from a manifest and hash-checked artifact ledger, failures are localized to explicit steps, and governance is enforced via HITL feasibility gates before tool/code execution and before final release. This safeguards-first design supports ICDS-style digital-society evidence workflows that must produce transparent outputs under real-world constraints on cost, latency, and risk.

A. Future Work

Our next steps focus on strengthening CHAINS along the dimensions that matter for trustworthy deployment and evaluation in digital society settings. First, we will expand controlled benchmarking under realistic constraints with repeated trials across a topic suite, matched-budget baselines, and distributional reporting of completion, cost/latency, and failure modes. Second, we will extend deterministic verification from structural checks to richer accountability checks, including stricter claim-to-evidence linking, citation/identifier validation, and policy-driven constraints on tool usage and external calls. Third, we will broaden server-side integrations, datasets, domain tools, and specialized or fine-tuned models while preserving governance semantics. Fourth, we will implement cross-episode learning with audit-preserving updates, including episodic memory, semantic memory over prior artifacts, and rubric-based patch artifacts. Finally, we will strengthen stakeholder-facing packaging for DSS use cases through standardized release bundles containing a primary deliverable, transparency summary, reproducible artifact pack, and verifier report.

B. HPC and Quantum-Enabled Extensions for CHAINS

Beyond software-only improvements, we propose two infrastructure upgrades—HPC-backed execution and quantum-enabled retrieval/optimization—to expand the scope of micro-experiments, improve evaluation rigor, and enable new classes of agentic workflows while preserving governance and auditability.

HPC-backed micro-experiments and scalable benchmarking. A key limitation of current agentic lab systems is that experiments are typically constrained to local resources, which biases evaluation toward small tasks and discourages repeated trials. We will integrate CHAINS with an HPC environment, such as a Slurm-based cluster, by introducing an execution adapter that converts approved experiment plans into batch jobs. Under this design, Experiment_RunnerAI does not directly execute heavy workloads; instead, after HITL approval it submits a job bundle consisting of a container or environment specification, a deterministic entrypoint script, and an input artifact manifest. The scheduler returns a job ID that is tracked in the run manifest, and all outputs—logs, metrics, plots—are collected into the artifact directory upon completion. This enables larger topic suites, more repetitions per topic, environment capture with fixed seeds, strict queue policies, and explicit failure localization for scheduler errors or resource exhaustion.

Quantum-enabled retrieval and optimization under budget constraints. We further propose exploring quantum and quantum-inspired components as optional, governance-controlled modules within CHAINS. The goal is not to claim universal speedups, but to provide a testable pathway for hybrid classical–quantum workflows where quantum resources are treated as scarce, auditable tools. Near-term integration targets include quantum-assisted retrieval/reranking, where candidate documents are mapped into an embedding space and quantum or quantum-inspired similarity scoring reranks top-k candidates, and quantum-inspired search for agent routing and experiment selection, where tool selection and experiment scheduling are modeled as constrained discrete optimization problems. Solver outputs will be treated as recommendations subject to HITL approval, and the manifest will record the objective, constraints, solver configuration, backend mode, shot count when applicable, and selected action set.

Adding HPC and quantum modules enables additional DSS-relevant metrics beyond Eq. (1). We summarize these dimensions with an extended composite score:

$$\mathrm{Score}_{\mathrm{HPCQ}} = \alpha\, w^{\top} m + (1-\alpha)\, v^{\top} r, \tag{2}$$

where $\alpha \in [0,1]$, $\sum_i w_i = 1$, $\sum_j v_j = 1$, and all weights are nonnegative. The vectors $\mathbf{m} = \big[\mathrm{WCR},\ \mathrm{ABI},\ \mathrm{GAS},\ \mathrm{HSE},\ \mathrm{RI}\big]^{\top}$ and $\mathbf{r} = \big[\mathrm{CE},\ \mathrm{EC},\ \mathrm{HMR},\ \mathrm{GL}\big]^{\top}$ capture core accountability metrics and resource/governance metrics, respectively. CE is Compute Efficiency, EC is an Energy/Carbon proxy, HMR is Hardware-Mode Robustness across local/HPC/quantum modes, and GL is Governance Load, defined as inverse-normalized human time and approvals per run. Overall, CHAINS is an architectural step toward reducing fragmentation in modern research practice by unifying orchestration, provenance, telemetry, verification, and governance in one operational pipeline.

ACKNOWLEDGMENTS

This material is based upon work supported by the National Science Foundation CISE Graduate Fellowships under Grant No. 2313998. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

REFERENCES

[1] T. S. Ashton, *The Industrial Revolution 1760–1830*. Oxford, U.K.: Oxford University Press, 1997.

[2] H. L. Dreyfus, *What Computers Still Can't Do: A Critique of Artificial Reason*. Cambridge, MA, USA: MIT Press, 1992.

[3] T. Hagendorff and K. Wezel, "15 challenges for AI: Or what AI (currently) can't do," *AI & Society*, vol. 35, no. 2, pp. 355–365, 2020.

[4] K. Crawford and R. Calo, "There is a blind spot in AI research," *Nature*, vol. 538, no. 7625, pp. 311–313, 2016, doi: 10.1038/538311a.

[5] Agents Laboratory, "Agents Laboratory—A+B demo," 2025. [Online]. Available: https://app.agents-lab-experiments-project.com/. Accessed: Nov. 27, 2025.

[6] OpenAI, "Agents SDK," OpenAI Platform Documentation, 2025. [Online]. Available: https://platform.openai.com/docs/guides/agents-sdk. Accessed: Nov. 27, 2025.

[7] Microsoft Azure, "Azure Virtual Machines," 2025. [Online]. Available: https://azure.microsoft.com/services/virtual-machines/. Accessed: Nov. 29, 2025.

[8] Microsoft Azure, "Azure Blob Storage," 2025. [Online]. Available: https://azure.microsoft.com/services/storage/blobs/. Accessed: Nov. 29, 2025.

[9] S. Colvin et al., "Pydantic: Data validation and settings management using Python type hints," 2025. [Online]. Available: https://github.com/pydantic/pydantic. Accessed: Nov. 27, 2025.

[10] J. Tang, L. Xia, Z. Li, and C. Huang, "AI-Researcher: Autonomous scientific innovation," *arXiv preprint arXiv:2505.18705*, 2025.

[11] Y. Yamada, R. T. Lange, C. Lu, S. Hu, C. Lu, J. Foerster, J. Clune, and D. Ha, "The AI Scientist-v2: Workshop-level automated scientific discovery via agentic tree search," *arXiv preprint arXiv:2504.08066*, 2025.

[12] C. Lu, C. Lu, R. T. Lange, J. Foerster, J. Clune, and D. Ha, "The AI Scientist: Towards fully automated open-ended scientific discovery," *arXiv preprint arXiv:2408.06292*, 2024.

[13] D. Castelvecchi, "Researchers built an 'AI Scientist'—what can it do?" *Nature*, vol. 633, no. 8029, pp. 266–266, 2024.

[14] J. Gottweis et al., "Towards an AI co-scientist," *arXiv preprint arXiv:2502.18864*, 2025.

[15] Q. Xie et al., "How far are AI scientists from changing the world?" *arXiv preprint arXiv:2507.23276*, 2025.

[16] X. Tang et al., "Risks of AI scientists: Prioritizing safeguarding over autonomy," *Nature Communications*, vol. 16, no. 1, Art. no. 8317, 2025.

[17] X. Tang et al., "Prioritizing safeguarding over autonomy: Risks of LLM agents for science," in *Proc. ICLR Workshop on Large Language Model (LLM) Agents*, 2024.

[18] W. Villalobos, "CHAINS: Collaborative Hybrid AI-Researcher Networked System," GitHub repository, 2025. [Online]. Available: https://github.com/WilbertFV/CHAINS. Accessed: Nov. 29, 2025.