MADS-CPS: A Machine-Checkable Admissibility Contract for AI Scientists in Autonomous Laboratories

Mateo H. Petel [1]

Abstract

AI scientists are moving from assistive tools toward systems that can generate hypotheses, orchestrate tools, interpret intermediate evidence, and in some domains close the loop with experimental execution. Existing evaluations increasingly measure capability, while release-relevant authorization in autonomous laboratories also requires evidence about one concrete run: whether its record is inspectable, replayable, integrity-preserving, and controlled at irreversible boundaries. We introduce *MADS-CPS* (Machine-checkable Admissibility for Discovery Systems in Cyber-Physical Science), a run-level admissibility contract that defines the missing evidence boundary between AI scientist capability and cyber-physical release. The contract specifies a declared assurance envelope, a conformance checker over required artifacts and replay status, verification modes for restricted auditability, and fail-closed PONR release gating. We instantiate the framework in a robot-centric autonomous-laboratory reference implementation and evaluate it through an eight-case conformance challenge corpus, a verification-mode admissibility matrix, an independent replay-link experiment, and a controller-matrix study spanning baseline and stressed regimes. In this proof-of-structure setting, the checker agrees with all injected evidence-fault labels, independent replay matches producer reports in the evaluated replay-link runs, and the controller matrix shows that evidence-layer admissibility can remain stable while operational productivity diverges under coordination stress.

Accepted at the ICML 2026 AI for Science Workshop.
[1]Stanford University. Correspondence to: Mateo H. Petel <mpetel@stanford.edu>.

Proceedings of the 43rd International Conference on Machine Learning, Seoul, South Korea. PMLR 306, 2026. Copyright 2026 by the author(s).

1. Introduction

AI scientists now span a broad spectrum, from systems that assist with literature synthesis or hypothesis suggestion to agents that can plan experiments, invoke tools, interpret observations, and decide whether a scientific workflow should proceed (Lu et al., 2024; Yamada et al., 2025; Gottweis et al., 2025). In chemistry and autonomous-laboratory settings, this spectrum already extends into tool orchestration and experimental execution (Boiko et al., 2023; MacLeod et al., 2022; Canty et al., 2025; Coley et al., 2019). As these systems move closer to closed-loop scientific action, the central question shifts from capability alone to release-relevant admissibility: whether one concrete *run* of an AI scientist workflow exposes enough evidence for audit, replay, and authorization.

That question is especially sharp in autonomous laboratories. A single run can couple planning, tool invocation, sensing, coordination, and irreversible decisions. Failures are then often systemic, extending beyond model-local errors. A planner can be scientifically capable yet still produce a run that is difficult to inspect, replay-resistant, or release-risky. Conversely, a workflow can appear locally sensible while remaining globally weak as an externally checkable scientific record.

Current progress in AI scientists makes this gap harder to ignore. Recent systems increasingly demonstrate the ability to generate ideas, use external tools, coordinate agents, and interact with laboratory or domain infrastructure (Gottweis et al., 2025; Boiko et al., 2023). At the same time, emerging benchmark efforts increasingly measure practical scientific capabilities beyond textbook reasoning (Laurent et al., 2024; Panigrahi et al., 2026). This progress leaves a distinct trust problem: the field needs criteria for deciding when a *particular run* of an AI scientist is sufficiently inspectable and replayable to support release-relevant decisions.

This paper addresses that gap. We use MADS-CPS to denote Machine-checkable Admissibility for Discovery Systems in Cyber-Physical Science, and introduce it as a machine-checkable release-boundary evidence contract for AI scientist workflows in cyber-physical scientific settings. MADS-CPS defines the missing evidence boundary between AI scientist capability and cyber-physical release: the point

at which a concrete run must expose enough structured evidence to be inspected, replayed, and authorized while treating controller internals as outside the trusted base. The framework is intentionally narrow. Certification, domain safety cases, controller prescription, and scientific-validity guarantees remain outside its scope. Instead, it defines a run-level admissibility contract: the minimum evidence obligations required before one workflow run might count as auditable, replayable, and eligible for release-relevant authorization under a declared assurance envelope.

The central design principle is separation of *capability* from *admissibility*. The controller can be centralized, decentralized, rule-based, learning-based, or hybrid. MADS-CPS is deliberately agnostic to that internal choice. What matters is whether the workflow emits an externally checkable evidence package inside a declared envelope. Conformance is therefore a property of interface-level artifacts and validation predicates in contrast to planner identity.

The paper makes four contributions. First, it defines a run-level admissibility contract for AI scientist workflows in autonomous laboratories in terms of a declared envelope, required artifacts, admissibility predicates, conformance tiers, verification modes, and PONR release semantics. Second, it formalizes controller-agnostic conformance at the assurance layer: the checker operates on emitted evidence in contrast to controller internals. Third, it introduces a restricted-auditability model that preserves structural integrity under redaction while making explicit which predicates are lost when payload content is hidden. Fourth, it provides a reproducible reference instantiation in a robot-centric autonomous laboratory and evaluates the framework across conformance, audit-mode, replay, and controller-matrix studies that distinguish invariance in easier regimes from meaningful divergence under harder coordination stress.

The contribution is intentionally narrow but strategically placed. MADS-CPS specifies the run-level evidence contract that makes larger questions of scientific validity, laboratory safety, and governance externally inspectable. That is the right abstraction for a reusable trust substrate: strong enough to gate release and support replay, yet agnostic to the internal controller family.

2. Problem Setting and Scope

We study AI scientist workflows that emit four artifact classes for each run:

(i) a structured execution trace,

(ii) an evaluation report,

(iii) an evidence bundle, and

(iv) a release manifest.

The core question is the following:

> What minimum machine-checkable evidence obligations allow a third party to determine whether one concrete autonomous-lab run is admissible for audit, replay, and release from artifact-level evidence alone?

The scope is intentionally constrained. We consider workflows that operate under a declared assurance envelope and emit the required artifacts above. We focus on machine-checkable predicates over those artifacts: presence, schema validity, replay status, integrity, and PONR coverage. We also consider restricted auditability, where traces can be redacted while preserving structural information.

A PONR action is a transition after which a scientific, physical, regulatory, or reputational consequence becomes costly or impractical to reverse. In autonomous-laboratory settings, representative PONR actions include committing a sample disposition, transferring a sample or result to the next physical workflow stage, and authorizing internal or public release of an experimental conclusion. MADS-CPS distinguishes the irreversible action from the PONR evidence obligation associated with it. A PONR obligation is the set of pre-release evidence conditions that must be witnessed before the gate can authorize the irreversible transition.

The reference organism is a robot-centric autonomous laboratory workflow, chosen because it is rich enough to involve coordination, tool use, and irreversible decisions while still supporting reproducible, run-level evaluation. Certification, regulatory compliance, bit-identical hardware replay, planner prescription, and universal payload disclosure remain outside the present claims.

3. Related Work and Positioning

MADS-CPS sits at the intersection of AI scientist systems, AI-for-science benchmarking, AI risk management, assurance-case structure, and regulated data integrity.

AI scientists and autonomous laboratories. Recent systems increasingly span literature-grounded ideation, agentic planning, tool orchestration, and autonomous experimentation (Lu et al., 2024; Yamada et al., 2025; Gottweis et al., 2025; Boiko et al., 2023). In parallel, self-driving laboratories and robotic scientific platforms have shown how closed-loop optimization and physical execution can accelerate materials and chemical discovery (MacLeod et al., 2022; Canty et al., 2025; Coley et al., 2019). Recent assessment work such as ADePT also makes the robotics side more explicit by characterizing autonomous laboratory capability in terms of adaptability, dexterity, perception, and task complexity (Salazar-Villacis & Benyahia, 2026). This literature

clarifies what AI scientist systems can perceive, decide, and execute. MADS-CPS contributes a complementary layer: it specifies what evidence must exist so that one concrete AI scientist run can be externally audited, replayed, and release-gated.

Benchmarks for scientific agents. The evaluation landscape is also maturing. LAB-Bench measures practical biology-research capabilities such as literature search, protocol planning, and data analysis beyond textbook-style science questions (Laurent et al., 2024). HeurekaBench argues that many existing scientific benchmarks still miss end-to-end co-scientist workflows that generate open-ended hypotheses (Panigrahi et al., 2026). These efforts move evaluation closer to real scientific work. Their primary question remains capability: can the system perform useful scientific tasks? MADS-CPS asks an orthogonal question: when should one concrete run count as admissible for audit, replay, and release?

Risk management and governance. Frameworks such as NIST AI RMF and related generative-AI governance profiles provide useful lifecycle structure for identifying, managing, and communicating AI risk (Tabassi, 2023; Autio et al., 2024). They are important for organizational process and governance posture. Their level of abstraction differs from the one pursued here. MADS-CPS is narrower and more operational: it asks what a third party can verify mechanically about one particular run, given a declared assurance envelope and a fixed artifact family.

Assurance cases and data integrity. Goal-based standards and structured assurance-case formalisms elevate evidence and argumentation as first-class objects (UL Standards & Engagement, 2023; Object Management Group, 2023; Kelly & Weaver, 2004). Scientific and laboratory environments also impose demanding expectations around record integrity, traceability, and auditability (OECD, 2021; U.S. Food and Drug Administration, 1997). MADS-CPS is motivated by these traditions, with a narrower role: to define a bounded, controller-agnostic, machine-checkable evidence object whose integrity, replay status, and release admissibility can be evaluated mechanically.

Provenance, research objects, and artifact integrity. MADS-CPS can also be viewed as a domain-specific admissibility layer over broader provenance and artifact-packaging traditions. W3C PROV models provenance through entities, activities, and agents, which maps naturally to traces, tool actions, and AI scientist controllers (World Wide Web Consortium, 2013). RO-Crate packages research objects and their metadata using JSON-LD, making it relevant for portable MADS-CPS evidence bundles (Soiland-Reyes et al., 2022). Workflow Run RO-Crate extends this idea to computational workflow execution provenance and aligns with W3C PROV, which is especially relevant for replayable AI-for-science workflows (Leo et al., 2024). SLSA frames provenance as verifiable information about how software artifacts were produced, providing a useful analogy for signed AI-scientist run artifacts (Open Source Security Foundation, 2025).

MADS-CPS therefore occupies a precise position. Capability benchmarks evaluate model or agent performance, full assurance cases organize system-level argument, runtime safety controllers enforce selected online constraints, and raw audit logs record chronological events. MADS-CPS supplies the release-boundary evidence object that these structures leave underspecified: a concrete, controller-agnostic artifact family whose schemas, integrity, replay status, and PONR evidence obligations can be checked for one run. Table 1 summarizes the resulting fault coverage and its limits.

Table 1. Fault coverage comparison. The final rows mark limits of the present layer: MADS-CPS detects evidence failures after instrumentation; uninstrumented actions and scientific truth remain outside its scope.

Fault or failure mode	Capability benchmark	Raw audit log	Runtime safety controller	MADS-CPS
Absent required artifact	outside	partial	outside	detects
Schema-invalid trace	outside	outside	outside	detects
State-hash or digest mismatch	outside	partial	outside	detects
Replay mismatch	outside	outside	outside	detects
Absent PONR witness	outside	partial	partial	detects
Stale or malformed release manifest	outside	outside	outside	detects
Safety-relevant action before instrumentation	outside	outside	possible	outside
Wrong scientific conclusion with valid evidence	possible	possible	possible	possible

4. MADS-CPS Model

The central object in MADS-CPS is a *run-level admissibility contract*. A run r is evaluated relative to a declared envelope

$$\mathcal{E} = (s, v, P),$$

where s is the scenario or profile identifier, v is the verification mode, and P is the set of release-relevant PONR evidence obligations for that scenario. The set P is profile-specific: it names the pre-release evidence obligations associated with transitions whose authorization requires witnesses. The run emits four required artifacts

$$A(r) = (T_r, R_r, B_r, M_r),$$

corresponding to a structured trace, an evaluation report, an evidence bundle, and a release manifest.

Verification modes. The mode $v \in \{\texttt{public}, \texttt{evaluator}, \texttt{regulator}\}$ specifies which predicates must be checkable and whether replay is mandatory. Public mode permits restricted auditability when payload content is redacted, while evaluator and regulator modes might require replay-capable traces.

Audit states under redaction. MADS-CPS distinguishes *full-admissible*, *restricted-admissible*, and *inadmissible* runs. A run is full-admissible when all predicates required by its declared mode, including replay and payload-dependent PONR checks when applicable, can be evaluated and pass. A run is restricted-admissible when structural and integrity predicates pass and authorized redaction removes payload content required for replay or payload-dependent PONR checks under public audit. Restricted admissibility is an audit status governed by redaction policy, separate from audience categories and physical-release permission. A run is inadmissible when a predicate required by its declared mode fails, or when predicate availability falls outside an allowed restricted-audit mode.

Replay contract. We use a bounded replay notion. *L0 replay* means that decision and enforcement outcomes are recomputable from the trace and recorded state transitions. *L1 replay* extends L0 by including recorded observations. MADS-CPS targets declared-fidelity replay, distinct from bit-identical hardware replay.

Definition 4.1 (Conformance checker)**.** *For a run r under envelope $\mathcal{E} = (s, v, P)$, define the checker $C(r; \mathcal{E})$ by*

$$\begin{aligned}
\mathrm{Tier1}(r) &\iff \textit{all artifacts in } A(r) \textit{ are present and schema-valid},\\
\mathrm{Tier2}(r) &\iff \mathrm{Tier1}(r) \wedge \mathrm{bundleOK}(B_r) \wedge \mathrm{replayOK}_v(r),\\
\mathrm{Tier3}(r) &\iff \mathrm{Tier2}(r) \wedge \forall p \in P,\ \mathrm{witnessed}(p, T_r).
\end{aligned}$$

Here $\mathrm{bundleOK}(B_r)$ *requires machine-checkable admissibility predicates recorded in the evidence bundle, including schema-validation status and replay status;* $\mathrm{replayOK}_v(r)$ *denotes replay success whenever replay is required by mode v; and* $\mathrm{witnessed}(p, T_r)$ *holds when T_r contains the required pre-release witness events for the PONR evidence obligation p, according to the declared profile.*

Definition 4.2 (Fail-closed release)**.** *A release-relevant transition is authorized only if the run satisfies the admissibility conditions required by its declared envelope. When those conditions remain unestablished, release is denied, with explicit auditable override available only as a recorded profile mechanism.*

Definition 4.3 (Release decision semantics)**.** *A checked run induces a release-boundary decision*

$$\Delta(r; \mathcal{E}) \in \{\mathsf{deny}, \mathsf{restricted}, \mathsf{authorize}\}.$$

The decision is authorize *when all predicates required by the declared envelope pass;* restricted *when structural and integrity predicates pass and an allowed redaction policy removes payload content needed for replay or payload-dependent PONR checks; and* deny *when a required predicate fails, a required artifact is missing, or predicate availability falls outside the declared mode. Only* authorize *permits execution past a physical PONR. Profiles might explicitly define narrower restricted-release classes.*

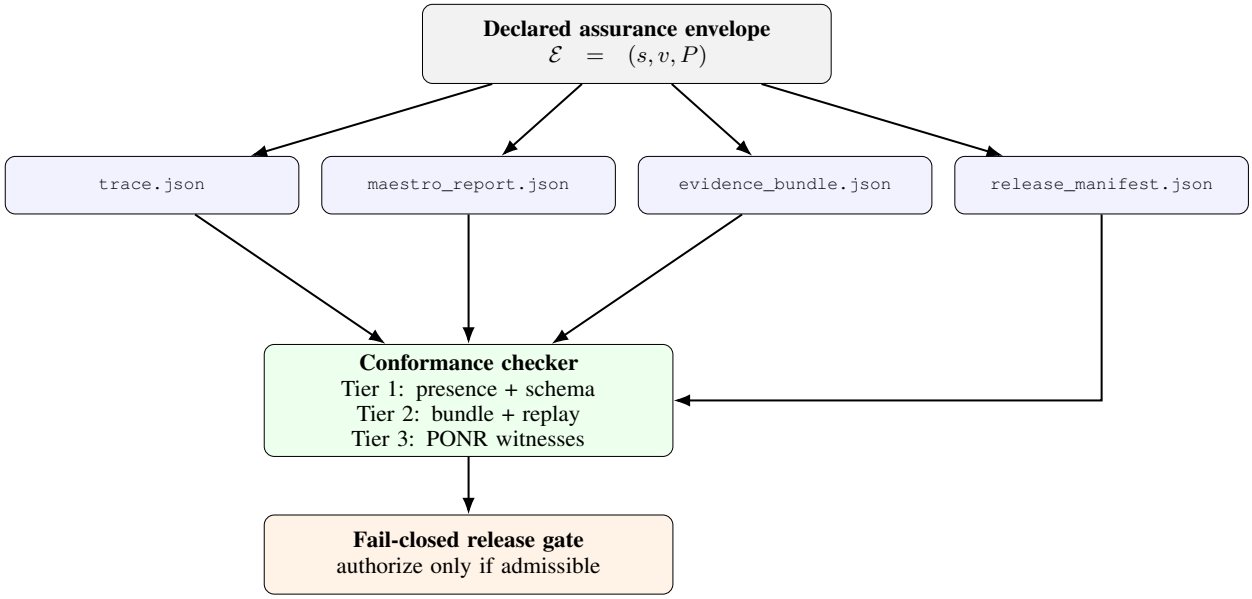

Figure 1. MADS-CPS evaluates each run relative to a declared assurance envelope and emitted artifact family. Admissibility is decided by a machine-checkable conformance checker and propagated to a fail-closed release boundary.

Table 2. Minimal MADS-CPS profile specification. A compliant profile may add domain-specific fields, while the checker must be able to evaluate these minimal obligations through artifact-level inspection alone.

Artifact	Minimal required fields	Checked predicates	Failure semantics
`trace`	`run_id`, `scenario_id`, `controller_id`, `events[]`, `state_hash_before/after`, `timestamps`, `ponr_events[]`	schema, hash chain, PONR witnesses, replay inputs	Absent artifact or structural fields fail T1; invalid hash or replay inputs fail T2; absent required PONR witnesses fail T3
`report`	`run_id`, `task_outcomes`, `safety_outcomes`, `replay_summary` `latency_summary`	report consistency, recomputation target	Absent artifact fails T1; inconsistency with replay target fails T2
`bundle`	`schema_ok`, `integrity_ok`, `replay_ok`, `mode`, `digest_map`, `checker_version`	bundle validity, replay status, integrity	Absent artifact fails T1; failed integrity or replay predicate fails T2
`manifest`	`release_id`, `run_id`, `envelope`, `authorized`, `override`, `signer`, `timestamp`	release decision, fail-closed semantics	Absent artifact or malformed release fields fail T1; unauthorized or unwitnessed PONR release fails T3

Profile authoring. A MADS-CPS profile must declare the scenario identifier, verification mode, required artifact schemas, replay fidelity level, PONR obligations, redaction policy, admissibility predicates, and release decision semantics. A profile is valid when artifact-level evidence enables an independent checker to determine Tier 1, Tier 2, and Tier 3 outcomes.

This formulation is intentionally interface-level. Conformance is computed over emitted evidence in contrast to planner internals, which allows the admissibility layer to survive controller substitution as long as the declared envelope and conformance-relevant artifact predicates are preserved.

4.1. Trusted Base and Failure Model

The checker assumes that schema definitions, hash functions, verifier implementation, artifact store, clock source, and signing or digest mechanism are inside the trusted base. It treats the AI scientist controller, generated hypotheses, scientific interpretations, and physical measurements as external to that trusted base. The threat model includes absent artifacts, malformed traces, replay inconsistencies, stale manifests, absent PONR witnesses, redaction-induced loss of replayability, and controller substitution. Fully compromised artifact generation, forged signatures by trusted keys, and physical failures outside the instrumented trace boundary remain outside the modeled failure class.

5. Properties of the Admissibility Layer

Proposition 5.1 (Tier monotonicity)**.** *For any run r under declared envelope $\mathcal{E}$,*

$$\mathrm{Tier3}(r) \implies \mathrm{Tier2}(r) \implies \mathrm{Tier1}(r).$$

Proposition 5.2 (Controller invariance of conformance)**.**

Fix a declared envelope $\mathcal{E}$. Let r_1 and r_2 be runs produced by possibly different controllers. If all predicates read by $C(\cdot;\mathcal{E})$ coincide on r_1 and r_2, then

$$C(r_1;\mathcal{E}) = C(r_2;\mathcal{E}).$$

Proposition 5.3 (Redaction preservation boundary)**.** *Under structure-preserving payload redaction, schema-validity and integrity predicates can remain checkable while replay and payload-dependent PONR predicates can require withheld payload content.*

Proof sketches. Tier monotonicity follows directly from the nesting of the tier predicates. Controller invariance follows because the checker reads only artifact-level predicates and branches exclusively on artifact-level evidence. The redaction boundary follows because structural fields and integrity digests can remain observable even when payload content needed for replay is withheld. Full proofs are deferred to the appendix.

6. Reference Instantiation

We instantiate MADS-CPS in a robot-centric autonomous laboratory profile. The reference implementation emits `trace.json`, `maestro_report.json`, `evidence_bundle.json`, and `release_manifest.json`.

Two professionalized evaluation profiles matter in the evaluation. The `artifact_profile` exercises artifact-level and replay-level reasoning. The `release_profile` includes the PONR-bearing task `disposition_commit`, making it the stronger witness for Tier 3 and release-gating semantics.

Example release decision. A run in `release_profile` reaches a proposed `disposition_commit`. The trace records the run identifier, scenario, controller, state hashes, tool events, observations, and PONR pre-release witness events. The evaluation report records task and safety outcomes. The evidence bundle records schema, integrity, and replay status. The release manifest records the requested release decision, signer, timestamp, and any override. The checker first verifies artifact presence and schemas, then validates digest consistency and replay status, then checks whether the `disposition_commit` obligation records the required witness events. A false `replay_ok` field fails Tier 2. An absent PONR witness fails Tier 3. When the required predicates pass, the release manifest can authorize the transition under the declared envelope.

7. Integration Path and Overhead

MADS-CPS is intended to sit at the boundary between an AI scientist controller and the laboratory systems that make a run consequential. In a realistic deployment, the controller remains responsible for planning and coordination, while an adapter layer emits the trace, evaluation report, evidence bundle, and release manifest in the declared schema. The checker then runs before any release-relevant transition, such as committing a disposition, publishing a result, transferring a sample to the next workflow stage, or authorizing a physical action past a declared PONR.

This integration path is deliberately conservative. For an existing autonomous-lab stack, the expected insertion points are the orchestration layer, the electronic laboratory notebook or laboratory information management system, the robot-control middleware, and the release or sign-off boundary. The framework leaves those systems in place and requires that they expose enough structured state, event, and obligation information for the four artifacts to be generated and independently checked.

The overhead has two components. The first is computational: schema validation, bundle checking, and verifier-side replay. In the reported E3 setting, verifier-side $p95$ latency remains below 61 ms across the two scenarios in the reference implementation. The second is operational: teams must define the assurance envelope, maintain schemas, preserve integrity digests, and record PONR witnesses. The present evidence supports low verifier-side overhead for the implemented slice, while full deployment overhead remains an empirical question for future hardware-integrated studies.

8. Experimental Design

We evaluate MADS-CPS through four experiments that target complementary properties of the admissibility substrate.

Metrics. Let V denote the independent verifier, T_i the trace for run i, R_i the reported evaluation summary, and $\mathcal{F}$ the declared comparison fields. The strong replay match rate is

$$\mathrm{SRM} = \frac{1}{N}\sum_{i=1}^{N} \mathbf{1}[V_{\mathcal{F}}(T_i) = R_{i,\mathcal{F}}]\,.$$

Raw conformance is $C(r;\mathcal{E})$ on the emitted artifacts. Normalized-interface conformance is $C(\nu(r);\mathcal{E})$, where ν canonicalizes controller-specific metadata fields such as controller identifiers, scheduling metadata, and benign ordering differences that leave checked admissibility predicates unchanged. Controller invariance is therefore reported in two forms: raw invariance on emitted artifacts and normalized-interface invariance after this declared canonicalization. In E4, a run is *productive* when it completes at least one declared scenario task while satisfying the admissibility predi-

cates required by its envelope. A run is *admissible nonproductive* when it satisfies the evidence and safety-relevant admissibility predicates while declining, deferring, or failing to execute productive tasks under the stress condition. Productive and admissible-nonproductive values are reported as fractions over seeds.

E1: Conformance challenge corpus. We construct an eight-case corpus of positive and fault-injected run directories, including absent artifacts, schema-invalid traces, state-hash corruption, bundle replay mismatch, absent PONR evidence, and stale release-manifest fields. The metric is checker agreement with the intended conformance outcome.

E2: Verification-mode admissibility matrix. We evaluate which predicates remain checkable under full and redacted audit conditions. The outputs are predicate-availability judgments for schema validation, integrity, replay, and PONR coverage across verification modes.

E3: Independent replay link. An independent verifier recomputes the evaluation report from the trace in a separate process from the producer pipeline. We run two scenarios with twenty seeds per scenario and report strong replay match rate and seed-level verifier-side latency summaries.

E4: Controller matrix under baseline and stressed regimes. We evaluate two controllers, `centralized` and `rep_cps`, across three scenarios and four regimes: `baseline`, `moderate`, `stress`, and `coordination_shock`. For each controller–scenario–regime combination, we report raw conformance, normalized-interface conformance, strong replay match rate, productive-success rate, admissible-nonproductive rate, verifier-side latency, and controller-pair diagnostics. The goal is to determine where evidence-layer invariance holds, where controller behavior diverges under stress, and whether the checker remains interpretable when that divergence appears.

9. Results

Taken together, the four result blocks establish decidability, mode-sensitive auditability, externally verifiable replay, and a regime-dependent controller story in which evidence-layer admissibility can remain stable even when productivity separates under stress. The results provide evidence for the contract and checker; deployment safety effects require future hardware-integrated studies.

E1 establishes mechanical decidability. Across all eight challenge cases in Table 3, the checker agrees with the intended outcome. This is a small but load-bearing result: admissibility is described and decided by an implemented procedure that catches faults at each intended layer of the evidence stack.

Table 3. E1 conformance challenge corpus.

Case	Fault injected	Expected	Observed
`valid_artifact`	clean run	T3 pass	T3 pass
`valid_release`	clean run	T3 pass	T3 pass
`absent_report`	absent `maestro_report.json`	T1 fail	T1 fail
`invalid_schema`	schema-invalid trace	T1 fail	T1 fail
`digest_mismatch`	corrupted `state_hash_after`	T2 fail	T2 fail
`replay_mismatch`	evidence bundle records `replay_ok=false`	T2 fail	T2 fail
`absent_ponr`	absent PONR witness	T3 fail	T3 fail
`stale_manifest`	absent `release_id`	T1 fail	T1 fail

Table 4. E2 verification-mode admissibility matrix.

Predicate	Full	Evaluator	Regulator	Public redacted
Schema validity	yes	yes	yes	yes
Integrity	yes	yes	yes	yes
Replay	yes	conditional	conditional	redacted
PONR coverage	yes	conditional	conditional	redacted

E2 clarifies the redaction boundary. Table 4 shows that redaction preserves some forms of auditability. Schema validity and integrity remain checkable, while replay and payload-dependent PONR predicates can fall outside public-redacted audit. This distinguishes restricted auditability from failed auditability.

Table 5. E3 replay-link under an independent verifier using strong replay semantics. Bracketed values denote reported 95% seed-bootstrap intervals for verifier-side $p95$ latency.

Scenario	Controller	Seeds	Replay match rate	Verifier $p95$ (ms)
`release_profile`	reference	20	1.00	50.37 [40.48, 60.27]
`artifact_profile`	reference	20	1.00	35.37 [27.11, 43.62]

E3 makes replay externally checkable. In E3, strong replay match rate, as defined above over the declared comparison fields, is 1.00 in both reported scenarios across twenty seeds per scenario, showing that the evaluation report can be recomputed by an independent verifier and treated as externally checkable evidence.

Table 6. Focused E4 controller divergence under harder coordination stress. Productive and admissible-nonproductive columns are fractions over seeds; mean $p95$ is verifier-side latency in ms.

Controller	Raw conf.	SRM	Productive	Adm. nonprod.	Mean tasks	Mean $p95$
`centralized`	1.00	1.00	0.55	0.00	3.30	147.25
`rep_cps`	1.00	1.00	0.00	1.00	0.00	0.00

E4 separates admissibility from productivity. The controller matrix yields a more precise picture than a simple "same checker, same results" story. In baseline settings, raw controller invariance holds, and normalized-interface invariance also holds in baseline and moderate settings. Under harder coordination stress, however, behavior separates: Table 6 shows that on `scheduling_profile`, `centralized` remains productive while `rep_cps` becomes admissible nonproductive, even though both remain raw-conformant and strong-replay-matching. Evidence-layer admissibility can therefore remain stable even when operational productivity diverges.

10. Discussion

The comparative value of the four-artifact release-boundary gate is that it defines the evidence boundary between AI scientist capability and cyber-physical release. It occupies a gap between capability evaluation, raw logging, runtime enforcement, and full assurance cases. Capability benchmarks can show that an AI scientist is useful; runtime controllers can block selected actions; assurance cases can organize system-level evidence. Each structure serves a different layer, while MADS-CPS supplies a controller-agnostic, machine-checkable answer to whether one concrete run exposes enough evidence to be inspected, replayed, and considered for release under a declared envelope.

The results support the paper's operative claim: in the present instantiation, a third party can determine whether a run is structurally complete, replay-admissible under its declared verification mode, and appropriately witnessed at release-relevant boundaries.

This matters for AI scientists because the field is moving from isolated prediction tools toward systems that increasingly resemble scientific actors. Once a system can plan, coordinate, and act across a closed-loop workflow, trust must extend beyond benchmark accuracy or local task success. The field therefore needs a release-boundary evidence contract for concrete runs.

Viewed as a stack, MADS-CPS sits between controller execution and institutional release: below it sit models, tools, planners, and lab middleware; above it sit assurance cases, governance processes, and regulatory or organizational sign-off. The controller-matrix results sharpen that distinction. MADS-CPS separates assurance-validity from operational productivity. In baseline settings, controller substitution can preserve evidence-layer conformance. Under harder coordination stress, controllers can diverge sharply in productivity while still producing admissible, replay-consistent evidence.

The restricted-auditability result is equally important. In realistic scientific settings, full trace disclosure could be blocked by intellectual property, safety, or regulatory constraints. The relevant question is therefore which predicates survive redaction and which predicates become hidden. MADS-CPS makes that boundary explicit through the distinction between full-admissible, restricted-admissible, and inadmissible runs: structural validity and integrity can remain externally checkable even when replay and payload-dependent predicates require payload content hidden by authorized redaction.

11. Scope Boundaries

The implementation remains a reference implementation with synthetic workloads. Physical hardware actuation, wet-lab failure measurement, and human-operator studies fall outside the present evaluation. The results should therefore be interpreted as a proof of structure, with field certification and real-world safety improvement reserved for future deployment studies. The replay contract is limited to declared fidelity, distinct from bit-identical hardware replay. The PONR mechanism is a release-time gate distinct from a live runtime safety controller. Controller-independence is claimed only in bounded form: evidence-layer invariance holds in baseline settings, normalized-interface invariance holds in baseline and moderate settings, and harder coordination stress can reveal meaningful controller divergence while preserving admissibility.

12. Conclusion

We introduced MADS-CPS, a machine-checkable admissibility contract for AI scientist workflows in autonomous laboratories. The framework defines a declared assurance envelope, a conformance checker over required artifacts and replay status, verification modes for restricted auditability, and fail-closed PONR release gating. In a reproducible reference instantiation, the resulting admissibility layer is mechanically decidable, externally verifiable under declared audit modes, and stable across controller substitutions in baseline settings, while harder coordination stress reveals that admissibility can remain intact even when productivity collapses.

The broader implication is that closed-loop AI scientists need a run-level evidence boundary between capability and release if they are to become inspectable, accountable scientific actors beyond opaque automation stacks. Models, planners, and orchestration layers will continue to change. What should remain stable is the boundary at which a run becomes admissible for replay, audit, and release.

Impact Statement

MADS-CPS is intended to support responsible deployment of AI scientist workflows by making release-boundary ev-

idence explicit, checkable, and auditable. In autonomous laboratories, structured traces, replay status, integrity evidence, and PONR witnesses can strengthen oversight, reproducibility, incident review, and accountable handoff between autonomous execution and institutional release. The principal misuse risk is evidentiary overreach: well-formed artifacts can be mistaken for certification, empirical scientific validation, or a complete safety case. MADS-CPS should therefore serve as a bounded admissibility layer that feeds domain safety analysis, governance review, and human responsibility structures.

Reproducibility Statement

The reported results are supported by an internal frozen artifact package containing the conformance corpus, redaction matrix, replay summaries, and controller-matrix exports used for E1–E4.

References

Autio, C., Schwartz, R., Dunietz, J., Jain, S., Stanley, M., Tabassi, E., Hall, P., and Roberts, K. Artificial intelligence risk management framework: Generative artificial intelligence profile. Technical Report NIST AI 600-1, National Institute of Standards and Technology, 2024. URL https://doi.org/10.6028/NIST.AI.600-1.

Boiko, D. A., MacKnight, R., Kline, B., and Gomes, G. Autonomous chemical research with large language models. *Nature*, 624:570–578, 2023. doi: 10.1038/s41586-023-06792-0. URL https://doi.org/10.1038/s41586-023-06792-0.

Canty, R. B., Bennett, J. A., Brown, K. A., Buonassisi, T., Kalinin, S. V., Kitchin, J. R., Maruyama, B., Moore, R. G., Schrier, J., Seifrid, M., Sun, S., Vegge, T., and Abolhasani, M. Science acceleration and accessibility with self-driving labs. *Nature Communications*, 16(1):3856, 2025. doi: 10.1038/s41467-025-59231-1. URL https://doi.org/10.1038/s41467-025-59231-1.

Coley, C. W., Thomas, D. A., Lummiss, J. A. M., Jaworski, J. N., Breen, C. P., Schultz, V., Hart, T., Fishman, J. S., Rogers, L., Gao, H., Hicklin, R. W., Plehiers, P. P., Byington, J., Piotti, J. S., Green, W. H., Hart, A. J., Jamison, T. F., and Jensen, K. F. A robotic platform for flow synthesis of organic compounds informed by AI planning. *Science*, 365(6453):eaax1566, 2019. doi: 10.1126/science.aax1566. URL https://doi.org/10.1126/science.aax1566.

Gottweis, J., Weng, W.-H., Daryin, A., Tu, T., Palepu, A., Sirkovic, P., Myaskovsky, A., Weissenberger, F., Rong, K., Tanno, R., Saab, K., Popovici, D., Blum, J., Zhang, F., Chou, K., Hassidim, A., Gokturk, B., Vahdat, A., Kohli, P., Matias, Y., Carroll, A., Kulkarni, K., Tomasev, N., Guan, Y., Dhillon, V., Vaishnav, E. D., Lee, B., Costa, T. R. D., Penadés, J. R., Peltz, G., Xu, Y., Pawlosky, A., Karthikesalingam, A., and Natarajan, V. Towards an AI co-scientist, 2025. URL https://arxiv.org/abs/2502.18864.

Kelly, T. and Weaver, R. The goal structuring notation—a safety argument notation. In *Proceedings of the International Conference on Dependable Systems and Networks Workshop on Assurance Cases*, 2004.

Laurent, J. M., Janizek, J. D., Ruzo, M., Hinks, M. M., Hammerling, M. J., Narayanan, S., Ponnapati, M., White, A. D., and Rodriques, S. G. LAB-Bench: Measuring capabilities of language models for biology research, 2024. URL https://arxiv.org/abs/2407.10362.

Leo, S., Crusoe, M. R., Rodríguez-Navas, L., Sirvent, R., Kanitz, A., De Geest, P., Wittner, R., Pireddu, L., Garijo, D., Fernández, J. M., Colonnelli, I., Gallo, M., Ohta, T., Suetake, H., Capella-Gutierrez, S., de Wit, R., Kinoshita, B. P., and Soiland-Reyes, S. Recording provenance of workflow runs with RO-Crate. *PLOS ONE*, 19(9):e0309210, 2024. doi: 10.1371/journal.pone.0309210. URL https://doi.org/10.1371/journal.pone.0309210.

Lu, C., Lu, C., Lange, R. T., Foerster, J., Clune, J., and Ha, D. The AI scientist: Towards fully automated open-ended scientific discovery, 2024. URL https://arxiv.org/abs/2408.06292.

MacLeod, B. P., Parlane, F. G. L., Rupnow, C. C., Dettelbach, K. E., Elliott, M. S., Morrissey, T. D., Haley, T. H., Proskurin, O., Rooney, M. B., Taherimakhsousi, N., Dvorak, D. J., Chiu, H. N., Waizenegger, C. E. B., Ocean, K., Mokhtari, M., and Berlinguette, C. P. A self-driving laboratory advances the pareto front for material properties. *Nature Communications*, 13(1):995, 2022. doi: 10.1038/s41467-022-28580-6. URL https://doi.org/10.1038/s41467-022-28580-6.

Object Management Group. Structured assurance case metamodel (SACM) specification, version 2.3. Specification, 2023. URL https://www.omg.org/spec/SACM/2.3.

OECD. GLP data integrity. Technical Report OECD Series on Principles of Good Laboratory Practice and Compliance Monitoring, No. 22, OECD Publishing, 2021. URL https://doi.org/10.1787/45779212-en.

Open Source Security Foundation. SLSA provenance. Specification, 2025. URL https://slsa.dev/spec/v1.2/provenance. Version 1.2.

Panigrahi, S. S., Videnović, J., and Brbić, M. Heurekabench: A benchmarking framework for AI co-scientist, 2026. URL https://arxiv.org/abs/2601.01678.

Salazar-Villacis, P. and Benyahia, B. The ADePT framework for assessing autonomous laboratory robotics. *Communications Chemistry*, 9(1):99, 2026. doi: 10.1038/s42004-026-01932-9. URL https://doi.org/10.1038/s42004-026-01932-9.

Soiland-Reyes, S., Sefton, P., Crosas, M., Castro, L. J., Coppens, F., Fernández, J. M., Garijo, D., Grüning, B., La Rosa, M., Leo, S., Ó Carragáin, E., Portier, M., Trisovic, A., Groth, P., Goble, C., et al. Packaging research artefacts with RO-Crate. *Data Science*, 5(2):97–138, 2022. doi: 10.3233/DS-210053. URL https://doi.org/10.3233/DS-210053.

Tabassi, E. Artificial intelligence risk management framework (AI RMF 1.0). Technical Report NIST AI 100-1, National Institute of Standards and Technology, 2023. URL https://doi.org/10.6028/NIST.AI.100-1.

UL Standards & Engagement. UL 4600: Standard for safety for the evaluation of autonomous products. Standard, 2023. URL https://www.shopulstandards.com/ProductDetail.aspx?productId=UL4600_3_S_20230317. Edition 3.

U.S. Food and Drug Administration. 21 CFR part 11—electronic records; electronic signatures. Code of Federal Regulations, 1997. URL https://www.ecfr.gov/current/title-21/chapter-I/subchapter-A/part-11.

World Wide Web Consortium. PROV-DM: The PROV data model. W3C Recommendation, 2013. URL https://www.w3.org/TR/prov-dm/.

Yamada, Y., Lange, R. T., Lu, C., Hu, S., Lu, C., Foerster, J., Clune, J., and Ha, D. The AI scientist-v2: Workshop-level automated scientific discovery via agentic tree search, 2025. URL https://arxiv.org/abs/2504.08066.