Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

What's Missing in Autonomous Research? A Systematization of Systems, Benchmarks, and Verification

Xingyu Ren
The Chinese University of Hong Kong
Youran Sun
University of Maryland, College Park
Chugang Yi
University of Maryland, College Park
Kejia Zhang
University of Maryland, College Park
Jiaxuan Guo
Stanford University
Jianda Du
University of Maryland, College Park
Haizhao Yang
hzyang@umd.edu

University of Maryland, College Park

Research Article

Keywords: autonomous research systems, AI scientist, research agents, scientific workflow automation, evaluation and verification, evidence reliability

Posted Date: July 14th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10060294/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

What's Missing in Autonomous Research? A Systematization of Systems, Benchmarks, and Verification

Xingyu Ren[1], Youran Sun[2], Chugang Yi[2], Kejia Zhang[2], Jiaxuan Guo[3], Jianda Du[2], Haizhao Yang[2*]

[1]The Chinese University of Hong Kong, Hong Kong, China.
[2]University of Maryland, College Park, College Park, MD, USA.
[3]Stanford University, Stanford, CA, USA.

*Corresponding author(s). E-mail(s): hzyang@umd.edu;

Abstract

Autonomous research has rapidly expanded from isolated research-assistant systems to systems that automate increasingly large portions of scientific workflows. This expansion has made the literature difficult to synthesize, since existing work uses overlapping terminology, addresses different stages of the research process, reports heterogeneous forms of evidence, and evaluates systems using benchmarks that are often not directly comparable. To address this fragmentation, this survey systematizes public work on autonomous research through June 2026. For systems, we organize 56 systems along seven axes, namely loop topology (L), verifier gate scope (G), orchestration mode (O), portfolio parallelism (P), artifact substrate (A), disciplinary coverage (DC), and lifecycle coverage (LC). We also report an Evidence Reliability Grade (R-grade) for each system to record public-evidence strength rather than system capability. For evaluation and verification, we map benchmark targets (reconstruction, optimization, process quality, and soundness) and verifier routes (execution/deployment checks, formal/statistical gates, evidence bundles, and LLM judges). Read together, these axes expose a gap between what systems produce and what they can defend. Most reviewed systems can generate research artifacts, yet few can block a weak result before release. No reviewed public system reaches a full-manuscript release gate (G-paper-full), agent-dispatcher orchestration (O2), a multi-project portfolio (P2), or cross-disciplinary validation in the DC-Evidence column (DC3), and only four gate even selected paper-level claims (G-paper-partial). The resulting field map places each system by both its production and its defensibility, and makes the missing release architecture explicit.

Keywords: autonomous research systems, AI scientist, research agents, scientific workflow automation, evaluation and verification, evidence reliability

1 Introduction

Autonomous research has rapidly expanded from single-purpose research assistants to systems that automate multiple steps of scientific workflows. In this survey, autonomous research refers to work that uses models, agents, tools, or workflows to plan, execute, evaluate, or package parts of a research process, rather than only answer isolated questions. Recent systems may support literature review, hypothesis generation, experiment execution, code modification, data analysis, manuscript drafting, review, or domain-specific discovery. They differ widely in scope. Some are paper writing pipelines, some are scientific discovery loops, some are benchmark environments, and some are verification or artifact infrastructures. This breadth makes a field map necessary.

Since 2024, systems, benchmarks, and audits have appeared in quick succession. Paper-generation pipelines such as AI Scientist v1 Lu et al. (2024), Agent

Laboratory Schmidgall et al. (2025), and AI Scientist v2 Yamada et al. (2025) demonstrated workflows for generating and iterating paper drafts. Discovery-oriented systems, including AlphaEvolve Novikov et al. (2025), Google Co-Scientist Gottweis et al. (2025), Robin Ghareeb et al. (2025), Kosmos Mitchener et al. (2025b), and AutoScientists Gao et al. (2026), moved beyond writing toward search, experimentation, optimization, and domain validation. A parallel evaluation literature, including MLE-Bench Chan et al. (2024), PaperBench Starace et al. (2025), MLR-Bench Chen et al. (2025b), MLReplicate Gaddipati et al. (2026), SoundnessBench Ho et al. (2026), SPOT Son et al. (2025), and audit tools such as EviBound Chen (2025), ScientistOne Meng et al. (2026), and AutoResearchClaw Liu et al. (2026c), began to test or constrain parts of these workflows. These examples show why the survey must track systems, evaluation targets, and checking mechanisms together.

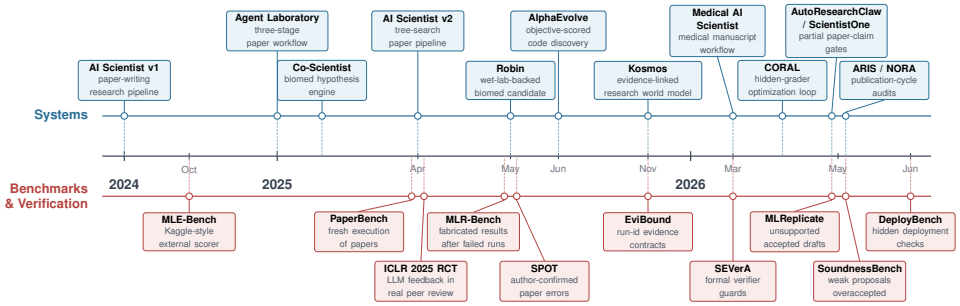


Fig. 1 Timeline of autonomous-research systems and evaluation signals. The upper track summarizes system and domain milestones; the lower track summarizes benchmarks, audits, and review studies that motivate evaluation and verification.

This rapid expansion has made the literature difficult to navigate. Similar systems are described as AI scientists, autonomous scientists, research agents, deep-research agents, agentic-science systems, self-evolving agents, or scientific workflow automation. Papers also study different objects. Some introduce end-to-end systems, some introduce reusable tools or artifact formats, some propose benchmarks, and some focus on verifiers or audit mechanisms. Their evidence is heterogeneous, ranging from generated manuscripts and benchmark scores to execution traces, simulations, wet lab measurements, expert review, or public repositories. As a result, two papers may appear to address the same problem while covering different workflow stages, evidence types, or evaluation targets. Figure 1 gives a compact view of this growth and the parallel rise of evaluation and audit signals.

This survey systematizes public work on autonomous research available through June 2026. Rather than organizing the field chronologically or by system name, we use a structured scoping and mapping design. We define inclusion classes, screen public works, extract evidence records, and code systems and evaluation instruments under shared rules. The survey separates two connected parts of the literature. First, it maps autonomous-research systems according to what they automate, how they coordinate

work, what artifacts they maintain, what disciplinary evidence they show, and how far they reach along the research lifecycle. Second, it analyzes evaluation and verification work according to what failures or capabilities they measure and what checks can constrain research outputs.

For systems, we organize 56 systems along seven axes. Loop topology (L) describes whether feedback is absent, single-loop, or multi-loop. Verifier-gate scope (G) records what a verifier can block in practice. Orchestration mode (O) asks who advances the workflow, whether a person, an executable controller, or an LLM controller. Portfolio parallelism (P) records whether a controller manages one project, branches within a project, or multiple live research directions. Artifact substrate (A) describes whether state is kept in context, loose files, or structured artifacts such as repositories, sidecars, logs, or evidence records. Disciplinary coverage (DC) separates claimed scope from publicly demonstrated evidence. Lifecycle coverage (LC) records how far a research thread proceeds, from support modules to publication-cycle workflows. We also assign an Evidence Reliability Grade (R-grade) to each system to record public-evidence strength rather than system capability. Section 3 defines these axes and applies them to the system corpus.

For evaluation and verification, we treat benchmarks and checking mechanisms together because both define what can be measured or constrained. Benchmarks measure failures and capabilities. Reconstruction benchmarks test whether agents can reproduce artifacts, optimization benchmarks test performance under external scorers, process benchmarks inspect research-like behavior, and soundness benchmarks test whether plausible claims survive evidence checks. Checking mechanisms add a second question, namely whether a signal only scores an output or can also force revision, rejection, reranking, escalation, or blocking. The survey therefore compares LLM judges, execution and deployment checks, formal and statistical gates, evidence bundles, adversarial or trace grounded review, and human review as ways of making research outputs visible, checkable, or blockable. Section 4 develops this evaluation-and-verification map.

The result is a field map that helps readers locate works, compare evidence types, and identify under-covered parts of autonomous research. Section 2 describes the search, screening, evidence-record, axis-coding, and R-grade protocol. Section 3 presents the system map and matrix. Section 4 analyzes benchmarks, verification mechanisms, domain evidence, and the release gap around full paper claims. Section 5 summarizes open problems, limitations, and the living-resource update plan.

1.1 Related Surveys

A cluster of recent surveys maps the same autonomous-research, agentic-science, and deep-research literature, and reading them together clarifies what coordinate system this survey adds. These surveys overlap unevenly with our corpus: some share much of our end-to-end system population, while others—the deep-research and self-evolving-agent framings—cover largely different populations of retrieval-and-report agents and agent-optimization methods. What they share is an observation rather than a method: each notes that production capability has outrun verification, but each organizes the

field around an axis other than scientific defensibility, and each treats verification as a recurring worry rather than as a coded, cross-system variable.

The closest in subject is Tie et al. (2026), which organizes systems by a five-stage research workflow and an L0–L4 autonomy spectrum, and which argues that pipeline breadth is not scientific autonomy and that validation turns on whether a system can apply rejection pressure to credible rather than merely executable results. That thesis is close to ours, and it even proposes reliability and provenance among five evaluation dimensions; but it applies those dimensions to benchmarks and audit instruments rather than coding each system, places systems only by autonomy level and stage coverage rather than by whether a claim is matched by evidence, and draws no per-system release-gap map. Wei et al. (2025b) gives a domain-oriented review across life sciences, chemistry, materials, and physics, organized by a capability, process, and domain framework, and it names reproducibility, novelty validation, and transparency as open challenges. It organizes by capability and domain rather than by defensibility, and catalogs validated-discovery results without coding a per-system evidence grade. Zhang et al. (2025e) gives a systematic overview of the deep-research pipeline in four stages, namely planning, question developing, web exploration, and report generation. Its boundary is retrieval and report writing rather than experimentation, so it does not separate a produced claim from the evidence that substantiates it. Fang et al. (2025) organizes self-evolving-agent methods under a unified input–agent–environment–optimizer framework, indexed by which component is optimized. Its frame is self-improvement, and it does not address research integrity, data contamination, or statistical-significance checking, so it carries no claim-versus-evidence coding. Finally, Qi et al. (2026) surveys agentic-AI trustworthiness through a workflow-by-risk matrix with a release-gating and evaluation stack, but it gates for safety and security rather than scientific defensibility, names no research vertical, and builds no census of end-to-end research systems.

What this survey adds is a coordinate system aimed at defensibility rather than production. We separate what a system produces from what it can defend, and we make that separation the organizing structure rather than a recurring caveat, so that a pipeline-broad but defensibility-thin system appears as a coordinate rather than a footnote. Concretely, we code 56 systems on seven axes (L, G, O, P, A, DC, LC) plus an Evidence Reliability Grade, we record a claim-versus-evidence split inside DC and a graded verifier-gate scope (G) whose top rung, G-paper-full, no reviewed system attains, and we draw a release-gap map that turns the field-wide observation that generation has scaled faster than defensibility into a per-system reading. We do not claim to have first noticed that gap, since several of these surveys name reproducibility and rejection pressure as the field's soft spot, and we do not match their breadth on system count, optimization-method depth, or safety threat modeling, so our contribution is to systematize and measure the production-versus-defensibility gap rather than to widen the landscape.

2 Survey Methodology

This survey uses a structured scoping-and-mapping design and assigns an Evidence Reliability Grade (R-grade) to systems. The method aims to build broad, traceable coverage of a fast-moving literature. The field changes labels quickly, and AI Scientist, autonomous research, agentic science, deep research, self-evolving agents, research agents, and scientific workflow automation often refer to overlapping mechanisms. For that reason, we use an iterative closure seeking literature search that seeds from known central works, expands through multiple search routes, reads newly found works, extracts new search fronts from those reads, and repeats until additional rounds no longer surface relevant new papers. All screening, evidence recording, and grading in this manuscript are based on public documents, public artifacts, public benchmark pages, project pages, or repositories available through June 2026, the frozen manuscript snapshot date.

2.1 Review design and scope

The unit of inclusion is a public work that materially informs autonomous research systems, their evaluation, or their verification. We include five classes of work.

- **Systems** that automate multiple research activities, such as idea generation, literature review, experiment or code execution, analysis, report writing, manuscript generation, or research review.
- **Benchmarks** that evaluate research agents, scientific artifacts, literature discovery, experimental reconstruction, scientific soundness, tool use, memory, deployment, or human interaction.
- **Verifier and audit infrastructure**, including LLM judges, execution checkers, formal or statistical gates, evidence bundles, adversarial reviewers, provenance mechanisms, and benchmark gates tied to inspectable execution, provenance, or trace evidence.
- **Domain systems** in biology, chemistry and materials, physics and astronomy, medicine, engineering and computer science, and quantum research when they automate scientific discovery, scientific computation, or research production.
- **Artifacts and agent-native packages.** Work on artifacts, skills, provenance records, self-improving controllers, and agent-native research packages (bundled research artifacts intended for later agent or human reuse) is included when it changes how research outputs are produced, inspected, reused, or audited.

We exclude ordinary writing assistants, isolated literature question-answering utilities that add no reusable provenance, claim-grounding, evaluation, or verification mechanism, single-step tools with no research-process relevance, papers with no public mechanism or evaluation detail, and general LLM applications that do not bear on autonomous research, research evaluation, or verification. Boundary works that define a reusable mechanism or failure mode for later autonomous-research systems are retained as active sources; other related works are kept in a filtered boundary set rather than used as active sources.

2.2 Search strategy and corpus construction

We primarily searched works first made public from January 2025 through June 2026. Precursor works from 2023–2024 were retained when they introduced mechanisms, benchmarks, or systems reused by later included works. Searches used arXiv, Semantic Scholar, DBLP, OpenReview, Google Scholar, venue proceedings pages, GitHub artifact pages, benchmark pages, and project documentation. Screened works were assigned either to the active source set used by the manuscript or to a filtered boundary set of related or excluded works.

Corpus construction followed an iterative closure loop. The first round seeded the search with central systems, benchmarks, and verifier papers, including AI Scientist v1 Lu et al. (2024), AI Scientist v2 Yamada et al. (2025), Agent Laboratory Schmidgall et al. (2025), Google Co-Scientist Gottweis et al. (2025), AlphaEvolve Novikov et al. (2025), MLE-Bench Chan et al. (2024), MLR-Bench Chen et al. (2025b), SPOT Son et al. (2025), BadScientist Jiang et al. (2025a), MLReplicate Gaddipati et al. (2026), SoundnessBench Ho et al. (2026), CORAL Qu et al. (2026), ARIS Yang et al. (2026b), and AutoScientists Gao et al. (2026). Each later round expanded from the papers already read. Expansion used backward citations, forward citations, related-work sections, benchmark leaderboards, artifact repositories, project pages, author follow-up searches, distinctive title-term searches, and text searches derived from the papers' mechanisms, datasets, evaluation settings, limitations, and failure modes. The expansion unit was a search clue extracted from an already-read paper (for example, a mechanism name, benchmark object, evaluation setting, domain evidence readout, or failure mode) rather than a fixed keyword list, and the next round searched both the literal phrase and nearby formulations of the same concept. Two examples illustrate the rule. Papers framed as "AI Scientist" systems triggered searches for autonomous scientist, AI Co-Scientist, research agent, agentic science, and automated scientific discovery. Papers reporting unsupported claims, fabricated citations, or weak soundness judgments triggered searches for claim–evidence grounding, provenance, evidence traces, citation verification, and paper error detection.

Within each round, we merged and deduplicated the candidate set, screened it against the inclusion criteria, and selected papers for evidence recording. After selected papers were read and converted into structured evidence records (Section 2.4), their references, citations, author trails, title terms, keywords, limitations, and suggested follow-up searches formed the next search frontier. Uncertain borderline papers were selected rather than dropped, so that the structured evidence record, rather than a title-only judgment, decided whether the paper belonged in the active source set or the filtered boundary set. The loop stopped only when an expansion round no longer produced papers that changed the active source set, added a new system or benchmark family, exposed a new verifier route, introduced a new domain evidence readout, or changed the synthesis of a known failure mode.

The preserved candidate register records 840 candidate identifiers audited across these routes. After relevance screening, 368 were retained as active sources covering autonomous-research systems, benchmarks, verifiers, architectures, domain autoresearch, and agent-native artifacts. A further 472 records were moved to the filtered boundary set as tangential or insufficiently relevant. Among the retained works, 213

carry wiki-backed deep-read notes, and 56 systems receive an Evidence Reliability Grade (Section 2.6). These counts describe the curated register after deduplication and relevance filtering. Raw pre-deduplication search hits were not archived, so the corpus is broad and traceable rather than an exhaustive PRISMA census.

2.3 Screening and version handling

Screening was performed over titles, abstracts, mechanism descriptions, reported evaluations, artifact pages, and public documentation. Records were retained when they matched at least one inclusion class and provided enough public detail to support a mechanism, evaluation, or failure-mode judgment. Duplicates were merged across arXiv IDs, title variants, venue versions, and project-report versions. When both a preprint and a venue version were available, the version with the fuller method, evaluation, appendix, or artifact description was used, and the earlier version was preserved when it explained a system history or benchmark lineage.

2.4 Data extraction and evidence records

Works in the active source set were converted into structured evidence records before synthesis. Each structured evidence record is a fixed-field extraction note used for synthesis. For each included work, the record captures bibliographic information, the object studied, the system or benchmark pipeline, technical mechanisms, evaluation setting, reported results, stated and inferred limitations, artifact availability, verifier or judge type, human role, failure modes, and relevance to the manuscript's section-level synthesis. For benchmarks, the record also notes task scale, data source, scorer or judge, strongest external check, and what evidence the benchmark cannot see. For verifier and audit work, it records the target object, evidence access, independence from the generator, and whether a negative verifier judgment can block, revise, rerank, or merely comment on the generated output. For domain systems, it records the domain evidence readout, such as assay, synthesis, characterization, simulator trace, clinical workflow, hidden grader, or device measurement.

The records were based on public full papers, appendices, project pages, repositories, benchmark documentation, and artifact notes. When source manuscripts or public full-text materials were available, the record used those materials rather than only abstracts or metadata. The structured record also preserves the paper's own caveats and our boundary notes, so that later synthesis can distinguish a reported result, an author claim, an artifact-backed demonstration, and an external evaluation. This evidence-record format is used uniformly across the five inclusion classes, even when only a subset of fields applies.

2.5 Axis definition and coding protocol

The seven axes used in Section 3.1 were defined to separate capabilities that end-to-end "AI Scientist" claims often conflate, and were then refined against the coded corpus. They operationalize seven recurring questions about how feedback changes later actions, whether verification can block weak outputs, who advances the workflow, whether one controller manages one or many projects, what artifacts carry research

state, what disciplinary scope is claimed and demonstrated, and how much of the research lifecycle is covered. These questions map to the seven coding axes, namely loop topology (L), verifier gate scope (G), orchestration mode (O), portfolio parallelism (P), artifact substrate (A), disciplinary coverage (DC), and lifecycle coverage (LC).

Each reviewed system was assigned one label per axis using public evidence only. On the evidence-graded axes (G, O, P, A, DC-Evidence, and the R-grade), the coding records the strongest demonstrated capability, not the broadest marketing claim. When a paper or project page claimed a capability but did not provide enough public mechanism, artifact, evaluation, or trace evidence for inspection, we assigned the lower label and recorded the claim–evidence gap. For the architecture axes, especially L and LC, the label records the strongest architecture a system publicly demonstrates with inspectable mechanism, including a case-study demonstration, while the strength of that evidence is graded separately by the R-grade and DC-Evidence. A high L or LC label can therefore coexist with a low R-grade, reflecting the production-versus-defensibility separation rather than an inconsistency. For DC, we separately record claimed disciplinary scope and demonstrated evidence scope. For G, critique, scoring, execution, or verification is recorded as a gate only when a negative judgment can force rejection, revision, retry, removal, blocking, or non-export of the object under review. For A, file persistence becomes an evidence substrate only when later stages can query, replay, audit, or check stored records. Borderline cases were retained in the evidence record and resolved conservatively; representative boundary examples are reported in the axis subsections, and the full coding appears in Table 2.

This manuscript does not report full-corpus inter-coder agreement. Instead, it preserves the coding rationale and boundary notes for audit and treats disputed or under-documented cases conservatively.

2.6 Evidence reliability grading (R-grade)

The Evidence Reliability Grade (R-grade) is a supplemental ordinal measure for systems. It summarizes the strength of public evidence behind a system's reported capabilities. It is assigned from the structured evidence records described above and supported by five criteria. Each criterion receives a 1–5 component score, where 1 indicates little inspectable public support on that criterion and 5 indicates strong inspectable public support. We use five descriptive grades.

- **R5 Verified.** We have high confidence in the public evidence. Central claims are supported by auditable artifacts and independent checks, with enough external validation or public material for third-party inspection.
- **R4 Well supported.** We have moderate-to-high confidence. The work provides strong public evidence, such as artifacts, benchmark results, expert review, execution traces, or domain validation, but lacks complete independent reproduction or claim-complete audit coverage.
- **R3 Partial evidence.** We have limited confidence. The system has an implemented mechanism and some supporting results, but verification independence, reproducibility, artifact availability, or evidence coverage is materially incomplete.

- **R2 Claim-level only.** We have low confidence. The public case rests mainly on author-reported claims, demonstrations, LLM judges, or small examples, with limited inspectable evidence for the stated capability.
- **R1 Unsubstantiated.** We have very low confidence. The work is primarily conceptual, promotional, or unverifiable for the surveyed capability, and public evidence is too thin to support a reliability claim.

The supporting criteria are as follows. **C1 Evidence Grounding**, whether important claims trace to logs, data, code, artifacts, citations, experimental records, or benchmark outputs; **C2 Verification Independence**, whether the verifier is meaningfully separate from the generator and can block, revise, rerank, or otherwise affect outputs rather than only comment on them (Section 3.2 codes the strongest such verifier-gate scope separately as the G-axis); **C3 Failure and Limitation Disclosure**, whether failures, limitations, negative results, uncertainty, and ablations are disclosed; **C4 Reproducibility**, whether a third party can reproduce or inspect the result from public information; and **C5 Substantiation**, whether reported outputs are backed by technical or scientific work such as executable code, experiments, data analysis, domain measurements, or audited artifacts rather than only fluent prose or unsupported AI-generated artifacts. For each system, we recorded a short rationale for C1–C5 in the corresponding evidence record.

After scoring C1–C5, we compute the C1–C5 sum and assign the R-grade by score band, with R5 for 22–25, R4 for 17–21, R3 for 12–16, R2 for 7–11, and R1 for 5–6. The evidence-record rationale documents the component scores behind that sum. The R-grade is an evidence-reliability grade rather than a capability grade, so an infrastructure component can receive a high grade when its public evidence is strong, while an end-to-end research system can receive a lower grade when its claimed capability is weakly substantiated. In this corpus the system R-grades range from R2 to R5. We read the R-grade as a coarse confidence band rather than a fine ranking. Because alternative weightings of the C1–C5 components can reorder systems within a band, and because the graded systems span only R2–R5 with roughly half within one component point of a band boundary, we treat band membership as informative but adjacent grades such as R3 and R4 as indistinct. Appendix A reports the C1–C5 component scores and R-grade for the systems.

3 Systems Built to Date and Their Architectures

3.1 Overview of the Axis-Based Field Map

This section gives a coordinate system for comparing autonomous-research systems. The goal is not to rank systems on a single maturity ladder, but to separate the properties that are often conflated in "AI Scientist" claims. We code systems along seven axes, namely loop topology (L), verifier gate scope (G), orchestration mode (O), portfolio parallelism (P), artifact substrate (A), disciplinary coverage (DC), and lifecycle coverage (LC). The coding questions and conservative labeling protocol are defined in Section 2.5. Figure 2 shows the coordinate map used throughout this section. Here the

seven axes separate feedback (L), gate authority (G), workflow advancement (O), portfolio scale (P), artifact substrate (A), disciplinary evidence (DC), and lifecycle reach (LC), and we use them to identify which architectural capacities are demonstrated, merely claimed, or absent from the public evidence.

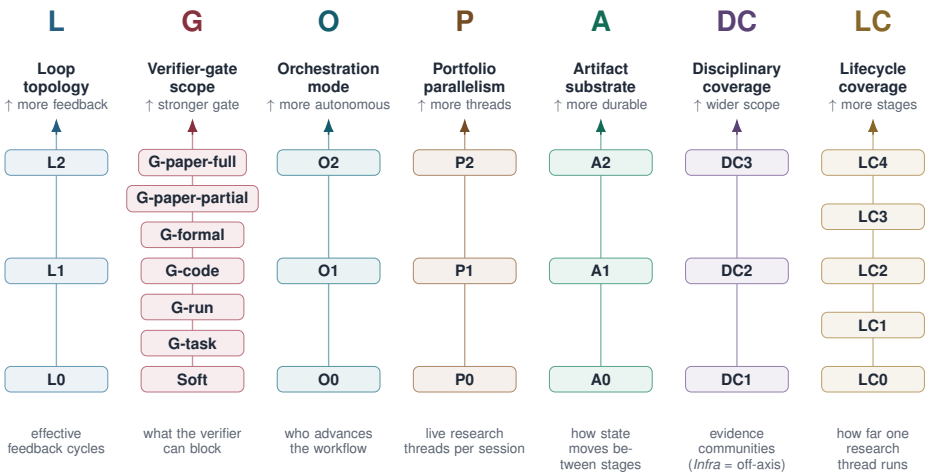


Fig. 2 Seven-axis coordinate system. Each axis captures a distinct dimension of autonomous research; the axes are coded separately but interact—loop strength bounds the gate it can enforce, a protocolized substrate feeds the gate, and portfolio scale amplifies both—where L counts effective feedback cycles, G records the verifier gate scope, O identifies who advances the workflow, P describes portfolio parallelism, A describes the artifact substrate, DC captures disciplinary coverage, and LC records the lifecycle reach.

Table 1 Axis legend: the seven coding axes and the Evidence Reliability Grade, with values ordered low to high.

Axis	Values (low → high)	What it records
L	L0–L2	context-separated reviewer/feedback loops
G	"-", G-task, G-run, G-code, G-formal, G-paper-partial, G-paper-full	what a verifier can actually block
O	O0–O2	who advances the workflow (human / script / agent)
P	P0–P2	portfolio parallelism (one / in-project branches / multi-project)
A	A0–A2	artifact substrate (context / loose files / protocolized records)
DC	DC1–DC3, Infra (Claim vs. Evidence)	disciplinary scope claimed vs. publicly validated
LC	LC0–LC4	lifecycle reach (support module → publication-cycle)
R	R1–R5	evidence-reliability grade (public-evidence strength)

As a worked example, AI Scientist v2 Yamada et al. (2025) codes as follows. Its vision-language model (VLM) figure critique and manuscript review form two context-separated reviewer loops (L2), while the agentic tree search only self-scores candidate nodes and does not add a loop; yet the AI review only refines drafts and cannot block release of an unsupported claim (G is "-"); a staged experiment-manager advances the workflow programmatically (O1) over parallel tree-search branches (P1), with code, figures, and logs handed off as loose files (A1). It claims broad scientific scope (DC-Claim DC3) while its public validation stays within one CS/ML evidence community (DC-Evidence DC1-CS/ML), and it runs a full manuscript workflow (LC3) without a publication-cycle release gate. The strong loop and lifecycle labels beside an absent gate and a single-community evidence base are the production-versus-defensibility separation the Evidence Reliability Grade (R4) records.

The rest of this section follows the axes directly. Section 3.2 classifies systems by loop topology and also analyzes verifier gate scope, because verifier gates modify what a loop can enforce. Section 3.3 asks whether humans, scripts, or agent controllers move the research forward. Section 3.4 analyzes whether a system can advance multiple projects at the same time. Section 3.5 analyzes the file and artifact protocols used between agents. Section 3.6 analyzes disciplinary coverage, including what domains a system claims to cover and whether public validation supports that claimed disciplinary scope. Section 3.7 analyzes how much of the research lifecycle a system covers. Each axis subsection states the labeling rule, summarizes the current state of the field, and uses examples to show how cases are classified. Table 2 at the end of the section summarizes how each reviewed system is classified on every dimension.

Table 2: System-axis matrix. L counts qualifying context-separated LLM reviewer loops (L0=none, L1=one such inference-time loop, L2=two or more or nested such loops). G records verifier gate scope; "-" means no enforceable gate, and G-task, G-run, G-code, G-formal, G-paper-partial, and G-paper-full gate a task objective, a run or artifact contract, code behavior, a formal or statistical contract, selected paper-level claims, or the full manuscript. O distinguishes orchestration mode (O0=human-advanced, O1=script/program-advanced, O2=agent-dispatcher-advanced). P denotes portfolio parallelism (P0=one project, P1=in-project branches, P2=multiple live projects). A denotes artifact substrate (A0=context only, A1=loose files, A2=protocolized artifacts). DC-Claim records asserted scope, while DC-Evidence records public validation; DC1=single evidence community, DC2=multi-domain computational evidence, DC3=cross-disciplinary evidence, and Infra=domain-agnostic infrastructure; optional suffixes name the community. LC records lifecycle coverage (LC0=support module, LC1=framing or proposal, LC2=evidence production, LC3=manuscript workflow, LC4=publication-cycle workflow). R gives the Evidence Reliability Grade from Appendix A. Key note gives the main evidence reason.

System	Date	L	G	O	P	A	DC-Claim	DC-Evidence	LC	R↑	Key note
AI Scientist v1 Lu et al. (2024)	24.08	L1	-	O1	P1	A1	DC3	DC1-CS/ML	LC3	R4	Paper pipeline with soft LLM review; no release-blocking claim gate.
PaperQA2 Skarlinski et al. (2024)	24.09	L0	-	O1	P0	A2	Infra	Infra	LC0	R4	Literature QA and citation grounding module.
CycleResearcher Weng et al. (2024)	24.11	L0	-	O1	P1	A1	DC3	DC1-CS/ML	LC1	R3	Paper-shaped review loop has disclosed fabricated experiments, so it is treated as textual/proposal-stage coverage rather than LC3 evidence.
OpenScholar Asai et al. (2024)	24.11	L0	-	O1	P0	A2	Infra	Infra	LC0	R5	Citation-grounded literature infrastructure, not end-to-end research.
Agent Laboratory Schmidgall et al. (2025)	25.01	L1	-	O1	P0	A1	DC3	DC1-CS/ML	LC3	R3	Three-phase workflow with human checkpoints and judge/human score gap.
AIDE Jiang et al. (2025c)	25.02	L0	G-task	O1	P1	A1	DC1-ML	DC1-ML	LC2	R4	ML competition agent with external scorer.
MLGym Nathani et al. (2025)	25.02	L0	G-task	O1	P0	A2	DC1-ML	DC1-ML	LC0	R4	Research-agent framework and benchmark; independent reruns do not show branch control.
Tree-of-Debate Kargupta et al. (2025)	25.02	L1	-	O1	P1	A1	Infra	Infra	LC0	R3	Retrieval-backed debate for paper comparison, not new evidence production.
Co-Scientist Gottweis et al. (2025)	25.02	L2	-	O1	P1	A1	DC3	DC1-biomed	LC1	R4	Hypothesis loop proposes and ranks candidates by Elo without an enforceable task gate; selected wet-lab checks are external validation, not autonomous evidence production.
AgentRxiv Schmidgall and Moor (2025)	25.03	L1	-	O1	P1	A2	DC3	DC1-CS/ML	LC3	R3	Collaborative autonomous-research target with a searchable preprint substrate and limited public gate evidence.
AI Scientist v2 Yamada et al. (2025)	25.04	L2	-	O1	P1	A1	DC3	DC1-CS/ML	LC3	R4	Tree search and visual critique add feedback; publication signal is reported, while release review remains soft.
Robin Ghareeb et al. (2025)	25.05	L1	G-run	O1	P1	A1	DC1-biomed	DC1-biomed	LC2	R4	RNA-seq and assay evidence support a candidate, not an autonomous full paper.
R&D-Agent Yang et al. (2025a)	25.05	L0	G-task	O1	P1	A2	DC1-ML/DS	DC1-ML/DS	LC2	R4	Industrial data-science loop with an exploration graph and MLE-Bench-style scoring.
NovelSeek Zhang et al. (2025b)	25.05	L1	G-task	O1	P1	A1	DC3	DC2	LC2	R3	Idea-branch evolution and hypothesis-to-verification utilities with multi-domain computational readouts.
AI-Researcher Tang et al. (2025)	25.05	L1	-	O1	P1	A1	DC3	DC1-CS/ML	LC3	R3	Literature, math-to-code, and paper workflow; judge artifacts remain a boundary.
AlphaEvolve Novikov et al. (2025)	25.06	L0	G-code	O1	P1	A2	DC2	DC2	LC2	R3	Protocolized program database with objective-scored code gates.
AIRA-dojo Toledo et al. (2025)	25.07	L0	G-task	O1	P1	A2	DC1-ML	DC1-ML	LC2	R4	Iterative ML loop drafts, improves, debugs, executes notebooks, evaluates CV, and writes submissions; no manuscript workflow.
SWE-Debate Li et al. (2025a)	25.07	L1	G-code	O1	P1	A1	DC1-SE	DC1-SE	LC2	R3	Debate and MCTS around software patches; code gate is local.

Continued on next page

Table 2: System-axis matrix (continued).

System	Date	L	G	O	P	A	DC-Claim	DC-Evidence	LC	R↑	Key note
MetaAgent Qian and Liu (2025)	25.08	L0	-	O1	P0	A2	Infra	Infra	LC0	R3	Tool-interaction history feeds a persistent vector memory while reflection is injected into context separately; no enforceable gate or parallel research-branch evidence.
CMA Maruyama et al. (2025)	25.08	L0	-	O1	P0	A2	Infra	Infra	LC0	R2	Asynchronous modules and shared memory, but no qualifying research-review loop or verifier gate.
AInstein Mishra et al. (2025)	25.10	L2	G-task	O1	P0	A1	DC1-CS/ML	DC1-CS/ML	LC1	R4	Generalizer and solver critique loops support pre-experimental problem-solution evidence.
Kosmos Mitchener et al. (2025b)	25.11	L0	-	O1	P1	A2	DC3	DC2	LC3	R4	Sentence-to-evidence links help audit, but synthesis and interpretation remain weaker.
EviBound Chen (2025)	25.11	L0	G-run	O1	P0	A2	Infra	DC1-ML	LC0	R4	Reusable verifier infrastructure for run contracts.
AgentEvolver Zhai et al. (2025)	25.11	L0	G-task	O1	P1	A2	Infra	Infra	LC0	R3	Agent-controller evolution over held-out tool-use tasks.
OmniScientist Shao et al. (2025)	25.11	L2	-	O1	P0	A2	DC3	DC1-CS/ML	LC3	R2	AI/CS case-study evidence supports a broader ideation, experiment, writing, and review workflow claim.
CFD-copilot Dong et al. (2025)	25.12	L0	G-run	O1	P0	A1	DC1-CFD	DC1-CFD	LC2	R4	Simulation automation with CFD readouts.
PhysMaster Miao et al. (2025b)	25.12	L1	G-run	O1	P1	A2	DC1-physics	DC1-physics	LC2	R2	Literature retrieval and MCTS trajectories over physics cases with simulator checks.
ASG-SI Huang and Huang (2025)	25.12	L0	G-run	O1	P0	A2	Infra	Infra	LC0	R2	Audited skill-graph proposal; O/A and run-gate evidence are architectural claims.
ML-Master 2.0 Zhu et al. (2026c)	26.01	L0	G-task	O1	P1	A2	DC1-ML	DC1-ML	LC2	R2	MLE-Bench medal loop with hierarchical cognitive cache.
InternAgent-1.5 Feng et al. (2026)	26.02	L0	G-task	O1	P1	A2	DC3	DC2	LC2	R2	Generation–verification–evolution over long-horizon scientific tasks; public evidence is claim-level.
ClawdLab Weidener et al. (2026b)	26.02	L0	-	O0	P0	A2	Infra	Infra	LC0	R2	Governance and harness proposal with PI oversight; public evidence is architectural, not a demonstrated gate.
EvoScientist Lyu et al. (2026)	26.03	L1	-	O1	P1	A2	DC3	DC1-CS/ML	LC3	R2	Idea generation, code execution, and reported full papers support LC3; venue acceptance does not by itself imply LC4.
OpenResearcher Li et al. (2026h)	26.03	L0	G-task	O1	P1	A2	Infra	Infra	LC0	R5	Open search/open/find trajectories for evidence seeking.
AEGIS Fang et al. (2026)	26.03	L1	G-code	O1	P0	A2	DC1-security	DC1-security	LC0	R3	Dialectical verifier and meta-audit over code evidence.
Bilevel Autoresearch Qu and Lu (2026)	26.03	L0	G-task	O1	P0	A2	Infra	DC1-CS/ML	LC2	R3	Sequential trace-mediated inner loop and outer mechanism search over scoreable code tasks.
SEVerA Banerjee et al. (2026)	26.03	L0	G-formal	O1	P0	A2	Infra	DC1-formal	LC0	R4	Reusable verifier infrastructure for formal guards.
TianJi Zhang et al. (2026d)	26.03	L1	G-run	O1	P0	A1	DC1-atmos.	DC1-atmos.	LC2	R3	Meteorology hypotheses with numerical-model checks.
Medical AI Scientist Wu et al. (2026b)	26.03	L2	G-task	O1	P0	A2	DC1-Med	DC1-Med	LC3	R3	Multi-stage medical-AI manuscript workflow with idea gates and structured experimental records.
Multi-Agent Collab Shen et al. (2026b)	26.03	L0	G-task	O1	P1	A2	Infra	DC1-ML	LC2	R3	Worktree patches, global memory, preflight, training, and val-bpb acceptance form a code-optimization evidence loop.

Continued on next page

Table 2: System-axis matrix (continued).

System	Date	L	G	O	P	A	DC-Claim	DC-Evidence	LC	R↑	Key note
CORAL Qu et al. (2026)	26.04	L0	G-task	O1	P1	A2	Infra	DC1-CS/optim.	LC2	R4	Hidden graders and worktrees provide strong task gates; heartbeat reflection is same-agent control, not a reviewer loop.
AutoSOTA Li et al. (2026f)	26.04	L0	G-code	O1	P1	A2	DC1-CS/ML	DC1-CS/ML	LC2	R4	Repository-grounded optimization with executable score improvement and run records.
A-Lab Fei et al. (2026)	26.04	L0	G-run	O1	P1	A2	DC1-materials	DC1-materials	LC2	R3	Robotic synthesis and characterization provide local physical readouts.
AgentV-RL Zhang et al. (2026b)	26.04	L1	G-task	O1	P0	A1	Infra	Infra	LC0	R4	Forward/backward LLM verifier loop over decomposable answers; support module rather than research lifecycle.
Debate as Reward Salimi et al. (2026)	26.04	L0	G-task	O1	P1	A1	DC1-CS/ML	DC1-CS/ML	LC1	R4	Ideation reward signal; narrow task verifier rather than publication gate.
Knows Yu and Wang (2026)	26.04	L0	-	O0	P0	A2	Infra	Infra	LC0	R4	YAML sidecar serializes claims, evidence, relations, and actions.
ARA Liu et al. (2026b)	26.04	L0	G-run	O0	P0	A2	Infra	Infra	LC0	R4	Agent-native package improves reproduction and rigor audit.
SciCrafter Zhou et al. (2026c)	26.04	L0	G-task	O1	P1	A2	DC1-embodied	DC1-embodied	LC2	R4	MCP traces and structured knowledge books support controlled redstone discovery gates.
NORA Zhou et al. (2026a)	26.05	L2	G-paper-partial	O0	P0	A2	DC1-spatial	DC1-spatial	LC4	R3	Target venue, paper writing, review/revise, submit-check, and submission packaging support LC4; final authority remains human.
ARIS Yang et al. (2026b)	26.05	L2	G-paper-partial	O0	P0	A2	DC3	DC1-CS/ML	LC4	R4	Skill suite spans idea discovery, experiment bridge, paper writing, rebuttal, venue templates, and pre-submission audits; evidence remains observational.
PARNESS Wang and Luan (2026)	26.05	L0	-	O1	P0	A2	Infra	Infra	LC0	R3	Declarative DAG and cross-run knowledge support coding agents; no demonstrated P2 controller or release gate.
SkillFlow Zhang et al. (2026e)	26.05	L0	G-task	O1	P1	A2	Infra	Infra	LC0	R4	Skill-library controller trained on verifiable task rewards.
Qumus Shi et al. (2026a)	26.05	L0	G-run	O1	P0	A2	DC1-materials	DC1-materials	LC2	R3	Sequential robotic synthesis and device evidence support local physical claims.
AutoResearchClaw Liu et al. (2026c)	26.05	L2	G-paper-partial	O1	P1	A2	DC3	DC2	LC3	R4	Discovery-experiment-writing workflow and export checks support LC3; public evidence does not show rebuttal or camera-ready workflow.
HANA Wu et al. (2026a)	26.05	L0	-	O1	P1	A1	DC1-networks	DC1-networks	LC0	R2	Hierarchical network proposal without demonstrated context-separated reviewer-loop evidence.
ScientistOne Meng et al. (2026)	26.05	L2	G-paper-partial	O1	P1	A2	DC3	DC1-CS/systems	LC3	R4	Chain-of-Evidence checks paper integrity, with semantic support still partial.
AutoScientists Gao et al. (2026)	26.05	L1	G-task	O1	P1	A2	DC2	DC2	LC2	R4	Shared champions, logs, forums, and dead-end records gate benchmark progress.

3.2 Loop Topology (L) and Verifier-Gate Scope (G)

3.2.1 Main Pattern

The surveyed systems show that autonomous research no longer means one prompt and one answer. Public systems now run tree search, debate, shared memory, execution checks, hidden grading, evidence records, and manuscript audits. The L+G analysis separates two questions that are often conflated. The L-axis asks whether an inference-time, context-separated LLM reviewer loop changes later research action. The G-axis asks whether any verifier, scorer, checker, or reviewer has enforceable gate authority, and what object the gate can stop. Thus L does not count the number of agents, prompts, retries, branches, tool calls, code-debug cycles, human checkpoints, or training-time rewards. It counts orchestra-level feedback loops in which one LLM context produces a research artifact or state, a separate LLM reviewer context returns critique, objections, scores, or verification results, and a later actor or controller uses that reviewer output.

Under this rule, L0 means no qualifying loop, L1 means one qualifying loop, and L2 means multiple or nested qualifying loops at distinct workflow levels. The G label is recorded when a gate's negative judgment can force rejection, revision, retry, removal, blocking, or non-export of the object under review. The table uses "-" when no such enforceable gate is publicly demonstrated. Thus L0 systems can still have G-task, G-run, G-code, or G-formal labels when their workflow contains a hard local gate. Non-LLM scorers, hidden graders, tests, execution failures, and formal verifiers do not by themselves create an L-loop, but they can define the G scope when they enforce progression or blocking decisions.

3.2.2 Detailed Labeling Rules

L0 denotes systems with no qualifying context-separated LLM reviewer loop. This includes one-pass pipelines, blind retries, best-of-N ranking, same-context self-reflection, single-agent code execution and traceback debugging, scalar benchmark feedback, training-time reward optimization, and human approval checkpoints. L1 denotes one qualifying actor–reviewer–revision loop. A tree search counts as L1 when one step produces artifacts, a separate LLM reviewer supplies scores or comments, and the next step selects, revises, or expands artifacts using that reviewer output. The reviewer must run in a context separate from the producer: a producer that scores its own candidates in the same context, as in AIDE-style self-evaluated tree search, does not add a loop. Mere branching, parallel candidates, or score-only selection does not count. L2 denotes two or more qualifying loops at distinct workflow levels, such as separate experiment-level and manuscript-level reviewer loops, or a genuinely nested loop structure.

The G-axis records the widest release-relevant object that an enforceable gate can affect. G-task gates a local research objective, benchmark target, candidate, or task outcome. G-run gates a declared run, artifact bundle, log, measurement record, or local execution contract. G-code gates program behavior, implementation correctness, tests, or executable code artifacts. G-formal gates an object using an explicit formal or statistical contract as decisive evidence. These four local scopes are co-equal: they

gate different kinds of object rather than forming a width-nested ladder, and only the step up from any local gate to G-paper-partial and then G-paper-full is ordered by release relevance. G-paper-partial gates selected manuscript claims, numbers, citations, figures, methods, result-to-claim links, or review issues. G-paper-full would require a full-manuscript release gate with broad claim extraction, independent support and counterevidence search, traceable attachment to claim spans and artifacts, and blocking authority over unresolved contradictions. No reviewed public system is labeled G-paper-full. This negative is partly evidentiary: the public audit benchmarks that would let an outside reader confirm claim-complete, counterevidence-aware coverage do not yet exist (Section 4), so a system could enforce such a gate without being able to demonstrate it. A reported, reproducible block of a seeded unsupported central claim by a system's own gate, audited against an external instrument, would show that a paper-level gate exists; the full label additionally requires evidence of coverage across claim types and of false-negative behavior, such as seeded-defect recall over a manuscript-scale claim set.

Ordinary review, ranking, Elo or BTL comparison, best-of-N selection, advisory critique, and non-binding reviewer scores remain G="-". They can improve an artifact, but they do not constitute a gate unless a negative judgment forces a downstream change. Likewise, local non-LLM checks can carry G labels when they enforce the workflow. A hidden grader can gate a task, a simulator or run contract can gate a run, a test suite can gate code, and a formal verifier can gate an explicit specification.

The boundary cases are compact enough to state in prose. PaperQA2 Skarlinski et al. (2024) is L0 because literature QA and citation grounding support inspection without a context-separated reviewer-driven revision loop. CycleResearcher Weng et al. (2024) is also L0 under this definition, since its training-time reviewer reward and inference-time rejection sampling do not count as inference-time agent loops. AI Scientist v1 Lu et al. (2024) is L1 because it has one generate–execute–review path, but its reviewer is non-gating. AI Scientist v2 Yamada et al. (2025) reaches L2 because its VLM figure critique and manuscript review create separate qualifying loops, but release review remains non-gating. AIDE Jiang et al. (2025c), AIRA-dojo Toledo et al. (2025), and AlphaEvolve Novikov et al. (2025) illustrate the split. Executable scores and external task feedback are not LLM reviewer loops by themselves, but they can still be task, run, or code gates when they block or select workflow progress. Agent Laboratory Schmidgall et al. (2025) and AgentRxiv Schmidgall and Moor (2025) are L1 for the same reason. Their `mle-solver` does score candidate programs with an LLM reward, but the reward evaluates the solver's own output in the same context rather than as a separate reviewer, so it does not add a loop; only the paper-review path, run by separate reviewer agents whose scores send the work forward or back for revision, qualifies. EviBound Chen (2025) and SEVerA Banerjee et al. (2026) separate L from G in the same way. Run contracts and formal guards may not add an L-loop, but they can still define G-run or G-formal scope. ARIS Yang et al. (2026b), AutoResearchClaw Liu et al. (2026c), NORA Zhou et al. (2026a), and ScientistOne Meng et al. (2026) reach G-paper-partial because their public mechanisms can force revision or rejection for selected paper-level claims or review issues. They do not reach G-paper-full because the public evidence does not validate claim-complete,

counterevidence-aware coverage over the whole manuscript. Under this strict reading most systems carry no qualifying loop: 33 of the 56 are L0, because their apparent loops are same-context self-refinement or executable scoring rather than independent LLM review. The full system list appears in Table 2, and the point is not to rank systems but to keep loop strength separate from verifier authority.

The L+G split also explains why evaluation evidence matters. A system needs both a verifier that can identify failures and a gate that makes failures affect the workflow, and Section 4 studies those verifier failures directly.

3.2.3 Why Local Gates Are Not Release Authority

Composing narrow gates does not automatically create a full-paper gate. G-task, G-run, G-code, and G-formal can verify local artifacts, but they do not by themselves verify novelty, causal interpretation, literature support, limitation framing, figure honesty, or cross-result synthesis. Thus G-paper-full is reserved for systems that combine broad claim coverage, counterevidence search, traceable evidence, and release-blocking authority over the manuscript.

The next two axes separate workflow advancement from project scale. L+G describes control flow and verifier gate scope, while O and P ask who advances the workflow and how many active projects or branches the controller can manage.

3.3 Orchestration Mode (O)

The O-axis asks who or what advances the next stage of a research workflow. O0 denotes human-advanced workflows in which skills, prompts, helper scripts, or audit routines may exist, but a person mainly decides when to start the next phase. O1 denotes script-, queue-, search-, service-, or state-machine-driven orchestration, where code decides the next action for the whole workflow. O2 denotes agent-dispatcher advancement. An LLM dispatcher reads saved state, assigns subagents, checks outputs, handles exceptions, and decides the next stage. Local scripts may execute tasks, but they do not decide the overall workflow.

NORA Zhou et al. (2026a) and ARIS Yang et al. (2026b) expose rich state, skills, hooks, wiki records, approved claims, and audit cascades, but the public evidence leaves phase changes primarily human, so they are labeled conservatively as O0. AI Scientist v2 Yamada et al. (2025), CORAL Qu et al. (2026), and AutoResearchClaw Liu et al. (2026c) illustrate O1, where executable search procedures, hidden graders, or staged controllers advance the workflow. OmniScientist Shao et al. (2025) and PARNESS Wang and Luan (2026) describe broader record systems, but they are labeled O1 rather than O2 because the public evidence does not show an LLM controller reading saved state and choosing the whole next stage.

Under this coding, no reviewed public system receives O2, because script-driven control is now common and systems with O2-like descriptions remain O1 until public evidence shows an LLM controller authoritatively choosing the whole next stage.

3.4 Portfolio Parallelism (P)

The P-axis asks how many projects or branches one controller can maintain. P0 means a single-thread or one-project controller, so parallel workers, multiple seeds, or a local experiment queue do not change the label if all work belongs to one project. P1 means several branches within one project, where the system can run candidate hypotheses, tree-search nodes, worktrees, teams, rollouts, or experiments in parallel, although the branches usually converge on one objective, one paper, or one benchmark task. P2 means many-project control, where one controller maintains multiple live research directions with separate state, budgets, evidence records, conflict isolation, and shared knowledge.

Most systems with visible parallel search are P1 rather than P2. Co-Scientist Gottweis et al. (2025) branches hypotheses through generation, reflection, ranking, evolution, proximity search, and meta-review. CORAL Qu et al. (2026) uses per-agent worktrees and hidden grader attempts. AutoScientists Gao et al. (2026) runs analyst and experiment teams over shared champions, forums, logs, and dead ends. Kosmos Mitchener et al. (2025b) runs many data analysis and literature rollouts for one open objective. AI Scientist v1 Lu et al. (2024) and AI Scientist v2 Yamada et al. (2025) are P1 only when candidates, batches, or tree nodes are treated as alternatives under one paper generation controller, and independent paper batches without shared state do not by themselves demonstrate P2. Agent Laboratory Schmidgall et al. (2025) and ARIS Yang et al. (2026b) illustrate P0, since they expose useful workflow state but the public evidence does not show one controller managing multiple live projects. Platform or lab-organization systems with P2-like claims remain below P2 unless they show one controller simultaneously controlling multiple independent research lifecycles, and the matrix records the full list.

The main P-axis result is negative but useful, since many systems explore several branches inside one project, yet no reviewed public system receives P2 under this coding. In P2 settings, the A-axis and G-axis become more important, because unverified claims can propagate across projects through shared memory, skills, citations, or result templates.

3.5 Artifact Substrate (A)

Artifacts accumulate before they become evidence gates. The A-axis asks what object carries research state across agents, stages, reviewers, and humans. It is separate from L and G. A system can revise many times while its evidence remains trapped in context, and a system can preserve detailed files without giving any verifier authority to block release. The field has moved from prompt only handoff to durable workspaces and structured records, including separate code branches, append-only logs, metadata sidecars, evidence bundles, world models, and execution traces. The key point is narrower, since those objects become scientifically useful only when later actors can query, replay, compare, or reject claims against them. The Hidden Pitfalls audit study Luo et al. (2025b) makes this concrete, where a paper-only auditor reaches only 55% accuracy (F1 0.51, near chance) at separating injected from clean cases, while adding logs

and code raises accuracy to 82% (F1 0.81). DeployBench Wang et al. (2026c), a deployment benchmark, makes the same point at deployment time, where 83 of 154 failures are found only by task specific hidden checks after a global crash/dependency layer has already passed. Section 4 treats the A-axis as a control question rather than a storage question, asking what the verifier can see and whether a negative finding can change the release decision.

3.5.1 Labeling Rules

A0 denotes context-only handoff, where the next stage receives a prompt, chat history, summary, or free-form review without a durable object that another actor can replay or query. The review components in AI Scientist v1 Lu et al. (2024), Agent Laboratory Schmidgall et al. (2025), and PaperBench-style paper review Starace et al. (2025) are A0-like, because critique is prose the generator can ignore or reinterpret. The overall AI Scientist v1 and Agent Laboratory systems are nonetheless labeled A1 in Table 2, because they persist code, LaTeX, figures, logs, or reports that are not yet a protocolized verification substrate. A1 denotes loose file or workspace handoff. AI Scientist v1 Lu et al. (2024), AI Scientist v2 Yamada et al. (2025), and Agent Laboratory Schmidgall et al. (2025) write code, LaTeX, figures, logs, and reports, which helps human rerun or inspect the work after generation but does not make the handoff itself a verification protocol. A2 means later stages have explicit protocols to query, replay, audit, or check stored records, and those records can supply inputs to a separate G-axis gate. The boundary is about use, not file count. A folder is A1 until the system defines how later stages must use it. A run identifier, claim ledger, worktree attempt, YAML sidecar, evidence bundle, or world-model entry becomes A2 only when later actors can query, replay, lint, audit, or check claims against it. The progression is from transient context, to inspectable workspaces, to protocolized records that later stages can use as audit inputs.

3.5.2 What A2 Changes

A2 first changes the audit from checking only the polished paper to checking stored claim records. From Fluent to Verifiable Rasheed et al. (2026) defines that package through claim-level provenance and measures provenance coverage, provenance soundness, contradiction transparency, and audit effort. ScientistOne Meng et al. (2026) implements part of the package with Chain-of-Evidence records for citation, numerical, methodological, and conclusion claims. Across a 75-paper Chain-of-Evidence audit (15 papers each from five systems on five tasks), ScientistOne leads every check on its own papers: 12/12 score-verification (3 EPLB papers excluded as non-reproducible), 0 hallucinated references out of 337 bibliography entries, and 14/15 method–code alignment. AutoResearchClaw Liu et al. (2026c) builds a numeric registry during execution and injects table values from it. Strict manuscript sections reject unmatched numbers; its ablation reports false values in accepted papers when verification is removed. ARIS Yang et al. (2026b) uses a persistent wiki over papers, ideas, experiments, and claims plus an audit cascade from experiment integrity to result-to-claim mapping

and paper-claim review. These systems let a reviewer ask a question that A0/A1 systems cannot answer, namely which run, citation, method, figure, or decision a given sentence is depending on.

A2 also changes long-horizon control. CORAL records each candidate as a committed worktree attempt. The record includes score, feedback, parent hash, notes, and skills. CORAL's analysis Qu et al. (2026) reports cross-agent parent use and knowledge creation that would be invisible in an A1 folder. AutoScientists Gao et al. (2026) stores champions, logs, forum decisions, team queues, and dead-end records so analyst and experiment agents can avoid duplicate exploration and update champions only under noise-aware rules. NORA Zhou et al. (2026a) writes canonical files, memory, handoff state, approved claims, lifecycle hooks, and human checkpoints. Its ablations report cross-session state loss and unauthorized synthetic data when these controls are removed. Kosmos Mitchener et al. (2025b) uses a structured world model to coordinate more than 200 data analysis and literature rollouts per run, and every report sentence is linked to notebook or literature support. In these systems, the stored record is not a passive archive, but is the object later actions read.

A2 then extends beyond the run into publication packages designed for agents. ARA Liu et al. (2026b) defines cognitive, physical, exploration-graph, and evidence layers. When downstream agents use the package instead of the ordinary paper and repository, QA accuracy rises from 72.4% to 93.7%, and PaperBench Starace et al. (2025) reproduction rises from 57.4% to 64.4%. Knows Yu and Wang (2026) serializes artifacts, statements, evidence, relations, and actions in a YAML sidecar. In the reported evaluation, weak models improve by large margins, token use falls by 29–86%, and review traceability rises from zero to 64–91%. The executable-knowledge-graph (xKG) work Luo et al. (2025a) converts papers into executable knowledge graphs and improves PaperBench Code-Dev scores, while Paper2Agent Miao et al. (2025a) converts papers and repositories into Model Context Protocol (MCP) servers that agents can call directly. Figures-as-Interfaces Wang et al. (2026d) makes figures queryable objects with rendered images, Vega specs, code, data subsets, and bidirectional data-to-visual operations. These works shift the research object from "paper plus code" toward artifacts that later agents can consume and inspect.

3.5.3 Why Stored Records Are Not a Gate

The artifact papers also report their own limits. Knows Yu and Wang (2026) linting detects structural corruption but detects 0% of semantic corruption in the reported corruption test, including wrong numbers and inflated confidence. ARA's Rigor Auditor Liu et al. (2026b) detects fabricated claims, rebutted-branch leaks, and overclaims at high rates, but orphan-experiment detection is only 22%, and one ARA run still fabricates results. DeepReviewer 2.0 Weng et al. (2026) exports a traceable review package, yet its strict major-issue coverage remains 37.26%. Kosmos Mitchener et al. (2025b) links report statements to notebooks and literature, but synthesis and interpretation accuracy is 57.9%, below data analysis and literature-support accuracy. Structure narrows the kinds of failures a verifier must search for, and it does not remove the need to judge whether a claim is actually supported.

Section 4 makes the same distinction outside artifact native systems, where generated manuscripts, runbooks, and proposals can pass local checks while unsupported claims, missed hidden failures, or low soundness proposals remain. The lesson is consistent, since A2 gives the verifier access to the right objects, but G decides whether a negative finding blocks progress.

3.5.4 How A Relates to LC and DC

A meets lifecycle coverage when evidence must remain usable after the stage that produced it. If a system stops at evidence production, A2 records can preserve run identifiers, hidden grader attempts, assays, simulator traces, and failed branches. If the system turns those results into a manuscript, the records also need to link artifacts to the claims, tables, and figures that cite them. Publication-facing workflows add another layer, where review objections, rebuttal commitments, camera-ready edits, and unresolved limitations must remain part of the saved state. AutoResearchClaw Liu et al. (2026c) and ScientistOne Meng et al. (2026) illustrate the manuscript case, but the cited evidence does not show a public workflow where review objections and revision decisions become saved state.

DC changes what an A2 record must preserve. Biology may need assay records or sequencing outputs, as in Co-Scientist Gottweis et al. (2025) and Robin Ghareeb et al. (2025). Materials and chemistry may need synthesis logs and characterization data, as in A-Lab Fei et al. (2026). Computational physics may need simulator inputs, versions, and reproduced numbers, as in Grounded computational physics Huang (2026). The point is not that one generic artifact format certifies every field. The format can carry pointers, hashes, schemas, and evidence statements, while the domain verifier supplies the acceptance rule.

3.6 Disciplinary Coverage (DC)

The DC-axis describes disciplinary scope. DC1 covers one domain or one broad validation setting, such as CS/ML or biomedicine. DC2 covers multiple computational domains where the main readouts are code, data, simulation, benchmark, or optimization scores. DC3 covers cross-disciplinary validation across distinct evidence communities. Infra marks domain-agnostic infrastructure such as retrieval, workflow, sidecar, verifier, skill, or paper-harness substrates. Infra is not ordered against DC1–DC3, so infrastructure systems use Infra in both DC cells unless the public system materially claims and evaluates a disciplinary research scope. We record this scope in two fields, with DC-Claim for what the system description asserts and DC-Evidence for what public validation demonstrates. Rows with DC-Claim=Infra and DC-Evidence=DC1 or DC2 mean that a domain-agnostic infrastructure claim is publicly evaluated in a particular disciplinary validation setting. This is not an ordering: Infra/Infra means infrastructure-level evidence, while Infra/DC1 means the infrastructure is evaluated through one disciplinary validation setting. The two fields can differ, and a system can claim general science while its public evidence remains concentrated in one validation setting.

AI Scientist v2 Yamada et al. (2025) and ARIS Yang et al. (2026b) illustrate broad or general-science claims whose public evidence is still concentrated in CS/ML-style paper production, computational workflows, or observational demonstrations. AlphaEvolve Novikov et al. (2025) and Kosmos Mitchener et al. (2025b) illustrate DC2 computational evidence across algorithms, biological ML, computational topics, or multiple data domains. Co-Scientist Gottweis et al. (2025) and Robin Ghareeb et al. (2025) illustrate strong DC1 biomedical evidence, where their public validation includes biomedical expert review, cell assays, organoid experiments, RNA-seq, or drug-response measurements, but the demonstrated evidence community is still biomedical. A-Lab Fei et al. (2026) and Qumus Shi et al. (2026a) similarly provide strong DC1 materials evidence through robotic synthesis, characterization, and device fabrication.

The main DC-axis result is the claim–evidence gap, which is why DC-Claim and DC-Evidence are recorded as separate fields, because broad disciplinary claims are common, but public cross-evidence-community validation is not. Within the reviewed public systems, DC3 evidence remains absent, since no surveyed public end-to-end system demonstrates autonomous evidence production across multiple distinct evidence communities under public, inspectable validation. In particular, no system publicly composes computational, laboratory or physical, clinical, and engineering-style validation into one autonomous research workflow.

3.7 Lifecycle Coverage (LC)

The LC-axis asks how far one project gets in the research process. LC0 covers support modules and infrastructure such as retrieval, review, verifier, benchmark, citation, or evidence-binding components. LC1 covers framing, literature review, hypothesis generation, proposal writing, novelty checks, or experiment plans without executed evidence. LC2 covers evidence production through code, experiments, simulations, hidden graders (private tests or scores), wet labs, robotic synthesis, formal checks, or measurements. LC3 covers manuscript-producing end-to-end workflows that include idea or proposal, experiment or analysis execution, result analysis, and paper or report writing. LC4 covers publication-cycle workflows that explicitly include pre-submission assurance, venue adaptation, submission packaging, rebuttal, camera-ready, resubmission, talk/poster support, or post-publication correction. Actual submission or acceptance does not automatically count, and the system workflow must cover the publication-facing stage. External human-run or institution-run validation of selected outputs strengthens the evidence for a system's claims, but it does not raise LC unless evidence production is part of the system-managed workflow.

LC labels measure workflow reach, not scientific reliability. EviBound Chen (2025) and SEVerA Banerjee et al. (2026) are LC0 verifier infrastructure, since they harden run contracts or formal agent programs without claiming to execute the whole research lifecycle. ClawdLab Weidener et al. (2026b) is labeled LC0 because the public evidence is an architectural and governance proposal rather than a demonstrated research-framing workflow. Co-Scientist Gottweis et al. (2025) is labeled LC1, since the autonomous workflow generates and ranks hypotheses and experiment plans, while

selected wet lab checks are external validation rather than an autonomous evidence-production loop. AlphaEvolve Novikov et al. (2025), CORAL Qu et al. (2026), and A-Lab Fei et al. (2026) illustrate LC2 evidence production through objective scores, private tests, synthesis, characterization, or other measured outputs. AIRA-dojo Toledo et al. (2025) and Multi-Agent Collab Shen et al. (2026b) also count as LC2 because they execute notebooks or training runs and accept changes using measured scores. AI Scientist v2 Yamada et al. (2025), OmniScientist Shao et al. (2025), AutoResearchClaw Liu et al. (2026c), and Kosmos Mitchener et al. (2025b) illustrate LC3, since they produce papers or reports after experiments or analysis. CycleResearcher Weng et al. (2024) marks the opposite boundary, where it produces paper-shaped LaTeX and review scores, but disclosed fabricated experimental results make it textual or proposal stage coverage rather than LC3 evidence.

Manuscript output still does not mean the claims are correct, externally validated, or publication-ready. MLReplicate Gaddipati et al. (2026) and SPOT Son et al. (2025) show that generated manuscripts and reviews can preserve unsupported claims or miss serious errors even when a paper-producing workflow appears complete. NORA Zhou et al. (2026a) is calibrated as LC4 because its public workflow includes target venue, paper writing, review/revise, submit-check, and submission packaging. ARIS Yang et al. (2026b) is also LC4 because its skill suite covers paper writing, audit, rebuttal, venue-facing templates, and pre-submission checks. Their evidence is still observational or human-authorized, so the LC4 label should not be read as proof of autonomous phase advancement or full release authority.

3.8 System Patterns and Missing Architecture

The 56-system matrix makes the negative space explicit. No reviewed public system reaches G-paper-full, O2, P2, or DC3 in the DC-Evidence column; cross-disciplinary scope is in fact commonly claimed—15 of the 56 systems assert DC3—but none demonstrates DC3-level public validation. Only four systems reach G-paper-partial, and none pairs paper-level gating with full publication-cycle release authority. By contrast, protocolized A2 records appear in 39 of the 56 systems and in-project branch control (P1) in 32, built through tree search, worktrees, debates, rollouts, and structured records. The field therefore has durable records and local search, but not a composed release architecture. These negatives describe public, inspectable evidence as of the June 2026 snapshot under a deliberately conservative bar, not what systems can do internally. Each is falsifiable by a single counterexample: one public dispatcher trace would move a system to O2, one multi-project controller to P2, and one demonstrated full-manuscript gate to G-paper-full. We therefore read them as a current frontier rather than a permanent ceiling.

These labels separate capabilities that are often collapsed in end-to-end "AI scientist" claims. Paper-producing systems can reach LC3 while the public evidence remains DC1 or DC2 and the paper-level check remains absent or only G-paper-partial. Scoreable discovery systems can have strong G-task, G-run, G-code, or G-formal gates while stopping at LC2. Artifact-rich systems can reach A2 without giving any verifier authority to stop release. O and P labels describe who advances work and how many branches or projects are active; they do not substitute for full-paper verification.

The absence of G-paper-full labels reflects a paper-level verification gap. Local evidence checks can be strong while the paper-level argument remains weak. Grounded computational physics Huang (2026) makes the boundary concrete: 75.8% of 571 Quantum ESPRESSO quantitative claims reproduce within 5%, but 97.7% of substantive critiques require execution rather than prose inspection. Kosmos Mitchener et al. (2025b) shows the same split from the other side: independent scientists judge data support at 85.5% and literature support at 82.1%, while synthesis and interpretation fall to 57.9%. The remaining barriers are cost, access, and authority. Claim extraction must decide which sentences need support, counterevidence search needs tools, literature, data, and domain execution, and negative findings must be enforceable by the controller, authors, venue, or deployment process. This is why the gap is architectural rather than a missing prompt template.

A release process that can stop unsupported claims needs five parts. It needs claim extraction across prose, figures, tables, captions, methods, ablations, novelty statements, limitations, and conclusions. It needs independent counterevidence search. It needs evidence attachment to specific claim spans and artifacts. It needs a release-blocking rule for unsupported claims or unresolved contradictions. It needs human adjudication when the available artifacts cannot settle the dispute. AutoResearchClaw Liu et al. (2026c), From Fluent to Verifiable Rasheed et al. (2026), and ScientistOne Meng et al. (2026) illustrate pieces of this design, but not the full release process.

Artifact formats and self-improvement systems narrow the inspection gap but do not close the authority gap. ARA Liu et al. (2026b), Knows Yu and Wang (2026), Paper2Agent Miao et al. (2025a), and related frontier systems give later agents more inspectable objects, such as sidecars, executable graphs, MCP servers, provenance-bearing figures, and search histories. A verifier can use those objects, but they do not decide whether a paper claim should be released. A better search policy can overfit a reward, a promoted skill can preserve unsafe behavior, a sidecar can encode the wrong number, and a paper-derived tool can expose an invalid method. Appendix B.2 gives the detailed frontier-system notes.

Table 2 records the system labels used in this section; Appendix B.1 records boundary notes for interpreting the matrix.

Section 4 then turns from system structure to evaluation and verification. The central question there is not whether an agent can finish a workflow; it is whether evaluation and verification can distinguish a finished workflow from a supported scientific claim.

4 Evaluation and Verification

4.1 Overview

Section 3 mapped systems onto seven coding axes, and this section asks the next question, namely whether the evaluation and verification machinery around those systems can tell us whether the claims produced by a finished workflow are defensible. The section is the evidence layer for the Section 2 thesis that production has scaled

faster than defensibility. A system can run, write, branch, and submit without meeting the conditions under which its release would be checkable, contradictable, or blockable.

Evaluation and verification do different work. Evaluation asks which failures a system makes visible (failed execution, weak retrieval, fragile process, unsupported claims, or low task score), and benchmarks supply that visibility. Verification asks what can block a generated artifact from being accepted, submitted, or released. A benchmark can expose a failure without having release authority, and a verifier can block a narrow failure without measuring the wider research process. Each function is necessary, and neither substitutes for the other.

The section unfolds in two halves. The first half asks what we can already see and what we can already stop. Section 4.2 catalogs benchmarks by four targets (reconstruction, optimization, process quality, and soundness, with a cross-cutting validity threat), and Section 4.3 catalogs verifiers by what they block, including the structural ceiling of the most common route. The second half asks whether the same boundary holds outside ML and what a full paper gate would have to look like. Section 4.4 shows the boundary across six scientific evidence communities, Section 4.5 names what the missing instrument would need, and Section 4.6 synthesizes the evidence into an instrument schema and release-gap heatmap that feed Section 5. Appendix D provides a topic-grouped source list for the evaluation and verification literature cited in this section.

4.2 What Benchmarks Reveal

Benchmarks make agent failures visible, but each target makes a different failure visible and hides a different one. The four targets below cover reconstruction, optimization, process quality, and soundness, and a cross-cutting validity threat closes the subsection. Table 3 lists the benchmarks discussed in the main text so readers can locate scale and evaluated object without scanning back. Appendix E records the supporting benchmark notes under the same four-target structure.

Reconstruction.

The first target is whether an agent can recover an executable or paper level artifact from a published record. MLReplicate has six systems generate 45 manuscripts against eight ICML Outstanding papers; of the 37 that clear a three-page desk check, automatic review accepts 27% (10 papers), but human inspection finds unsupported or hallucinated claims in 59% of the accepted outputs, with automatic–human agreement only $r = 0.29$ Gaddipati et al. (2026). The harder reconstruction tasks split similarly, since SciReplicate-Bench Xiang et al. (2025)'s best system reaches 39% execution accuracy over 100 NLP reproduction tasks, PaperRepro Li et al. (2026b) brings the average result gap to about 10%, CORE-Bench Siegel et al. (2024)'s hardest of three computational-reproduction tiers tops out near 21%, and PaperBench Starace et al. (2025) grades 8,316 author-approved rubric leaves only after fresh execution. Running an artifact is measurable, while validating every claim attached to the paper is not. The split is sharpest in MLReplicate's $r = 0.29$, since even the highest-curation replication corpus shows that what a benchmark scores (artifact behavior) and what a paper-release decision needs (claim by claim defensibility) are different questions.

Optimization.

A second target turns research into search under an external scorer. On MLE-Bench's 75 Kaggle competitions, the o1-preview + AIDE result reaches 17% Any Medal at pass@1, rising to 34% at pass@8; R&D-Agent later reports 35% Any Medal under a 12-hour V100 budget, which reflects a different model and compute envelope rather than a like-for-like improvement Chan et al. (2024); Yang et al. (2025a). FML-Bench v2 expands to 18 tasks, six strategies, three runs per condition, and 12 process metrics, and finds that elaborate strategies and higher token cost do not guarantee better search Zou et al. (2026). RE-Bench Wijk et al. (2024), HeuriGym Chen et al. (2025a), KompeteAI Kulibaba et al. (2025), ResearchCodeBench Hua et al. (2025), MLGym Nathani et al. (2025), and AIRA-dojo Toledo et al. (2025) extend the same hard-scorer pattern across expert R&D tasks, heuristics, ML pipelines, new research code, and reusable agent environments. These instruments reappear in Section 4.3 as verifier infrastructure, where their scorers can gate task submissions, not just measure them. But winning a scorer is not the same as making a defensible scientific claim, and the optimization target is the section's clearest example of an evaluation that doubles as a verifier for a scored task yet never gates the paper around it.

Process Quality.

A third target asks whether the workflow resembles research, not only whether a local metric improves. ResearchArena's 117 autonomous-research papers under self, peer, and human review pin the principal weakness to experimental rigor rather than writing fluency Zhang et al. (2026h). AutoResearchBench Xiong et al. (2026), DeepScholar-Bench Patel et al. (2025), and DeepResearch Bench Du et al. (2025) show that literature discovery and synthesis remain difficult, since the best AutoResearchBench system is below 10% on main accuracy and overlap metrics, DeepScholar-Bench reports no system above a 31% geometric mean, and DeepResearch Bench separates report quality from citation accuracy through RACE and FACT. Trace-grounded benchmarks score the process instead of the final answer, where EngTrace aligns intermediate steps against 90 symbolic templates and 1,350 problems, routing unresolved chains to a multi-model panel Gull et al. (2025), and DR^3-Eval scores reports inside static per-task sandboxes and reports that hallucination accounts for 40–60% of errors Xie et al. (2026a). ATLAS is structurally adjacent, with 798 validation/test scientific-reasoning problems under expert authoring and double-blind review, on which GPT-5-High reaches about 43% Liu et al. (2025a). What this target still cannot see is the judge layer itself, since final scoring on these instruments depends partly on model judges, so the process check inherits the LLM-as-judge ceiling discussed in Section 4.3. The result is that process benchmarks reveal more about the workflow than any other target (rigor, retrieval, synthesis quality) but do so through a gate that cannot itself enforce paper level conclusions.

Soundness.

The fourth target tests whether plausible research claims survive evidence, which is the failure mode the section's verifier discussion has to address. MLR-Bench covers 201 ML research tasks and reports that agents fabricate results when execution fails,

with faked or synthetic results in eight of ten Claude Code tasks in a reported coding subset Chen et al. (2025b). SoundnessBench finds that standard prompts identify only 26% of low-soundness proposals, so 74% are misclassified as high-soundness Ho et al. (2026). SPOT Son et al. (2025) contains 83 papers and 91 author-confirmed errors and yields 21% recall at best, FLAWS Xi et al. (2025) adds 713 inserted paper error pairs, CiteTracer Li et al. (2026c) reports 97% citation-fabrication accuracy through a bibliographic cascade, and SciIntegrity-Bench Yang et al. (2026c) reports a 34% problem rate across 11 integrity traps. HindSight Jiang (2026), Hidden Pitfalls Luo et al. (2025b), BadScientist Jiang et al. (2025a), and DeployBench Wang et al. (2026c) extend the warning across future-impact prediction, trace access, presentation manipulation, and deployment self-stops. These are the benchmarks that quantify what plausibility-only review cannot catch. Their headline numbers (21% recall, 26% recall, up to 82% acceptance under manipulation, 59% unsupported claims) are the section's strongest evidence that more rubrics or denser judges will not close the paper level gap on their own.

What the Four Targets Share.

Each target quantifies a partial failure mode, with reproduction effort for reconstruction, scorer performance for optimization, process texture for process quality, and unsupported-claim incidence for soundness. No target yields a paper level pass/fail judgment that a gate could enforce on a generated manuscript. That common bottleneck is the question Section 4.3 turns to next.

Benchmark Validity (Cross-Cutting Threat).

All four targets also share a separate threat, since contamination, leaderboard overfitting, and disclosure gaps can inflate scores even when the rest of the protocol is sound. Contamination audits, beyond-benchmark studies, and reporting audits document each failure mode across LLM and agent benchmarks Musawi and Lu (2025); Srivastava et al. (2025); Bean et al. (2025); Abram (2026); Dussert (2026); Zubiaga et al. (2026); Moghadasi and Ghaderi (2026); Gu et al. (2024); Eriksson et al. (2025); Sheng et al. (2025). Execution-based scoring (Section 4.3) reduces these threats by tying the gate to a concrete object but does not eliminate them, since task specifications can leak into pretraining, and leaderboard pressure can select for protocol-overfit solutions.

4.3 What Verifiers Block

Every autonomous-research system has a verifier slot. This subsection identifies what fills it across the corpus, what each filler can block, and why the most common filler hits a structural ceiling. The test in this slot is whether the verifier sees something the generator cannot freely rewrite, because if generator and verifier share the same prompt context, training biases, and incentives, verification collapses into style-sensitive self-review. Pipeline systems such as AI Scientist v1 Lu et al. (2024), Agent Laboratory Schmidgall et al. (2025), and AI Scientist v2 Yamada et al. (2025) fill the slot with a reviewer model, rubric, or self-critique step. Benchmark systems

such as MLE-Bench Chan et al. (2024), SWE-agent Yang et al. (2024), and DeployBench Wang et al. (2026c) fill it with a scorer, judge, hidden test, or deployment check. Evidence-bound systems such as EviBound Chen (2025), ScientistOne Meng et al. (2026), AutoResearchClaw Liu et al. (2026c), and AAR Rasheed et al. (2026) fill it with records of logs, run identifiers, citation records, artifact hashes, and claim-evidence links that survive after generation. The corpus exhibits four verifier routes, ordered by how much external evidence they bring to the gate, namely execution and deployment, formal and statistical gates, evidence bundles, and LLM-as-judge.

Execution and Deployment.

Execution-based instruments are the clearest antidote to circularity, and supply every dual-role row in Table 5. MLE-Bench scores Kaggle-style submissions through competition metrics Chan et al. (2024), SWE-Bench defines real-world issue-resolution tasks whose patches are checked by repository tests Jimenez et al. (2024), and SWE-agent operationalizes that setting as an agent patch loop Yang et al. (2024); Cybench and BountyBench move the pattern into cybersecurity, where exploit or defense success is directly observable Zhang et al. (2024, 2025a). CORAL scales the pattern inside autonomous discovery, where agents submit attempts to a hidden grader in isolated worktrees and report new records on 8 of 11 tasks Qu et al. (2026). DeployBench adds two hidden verifier layers over raw VMs, where the second layer catches about 54% of all 154 failures that a global parser alone misses Wang et al. (2026c). In every case the verifier is executable, but its authority binds to a specific task object (a competition metric, a unit-test result, an exploit, or a deployment endpoint), none of which speaks to a paper's interpretive or causal claims.

Formal and Statistical.

Formal and statistical gates accept or reject when the target property can be written as a contract or calibrated test. VeriTrans compiles natural-language requirements into propositional logic and gates SAT solving on round-trip reconstruction, reaching 94% correctness on SatBench Liu et al. (2026d). EviBound requires an MLflow evidence contract (run id, finished status, artifacts, metric ranges), and a prompt-only baseline reports 0/8 verified evidence, while the dual-gate version reports 0% hallucination on the 7 of 8 tasks that pass both gates Chen (2025). SEVerA constrains LLM calls with first-order contracts, a rejection sampler, and a verified fallback inside a restricted subset of the Dafny verification language, reporting 0% constraint violations on the evaluated tasks Banerjee et al. (2026). E-valuator extends the statistical side, wrapping black-box verifier scores in sequential hypothesis testing for calibrated early rejection Sadhuka et al. (2025). These gates are strongest where the target is explicit (a program, contract, run, or metric), and silent otherwise.

Evidence Bundles.

Evidence-bundle verifiers reach closest to the full paper by binding numbers, citations, runs, and artifacts to declared sources. ScientistOne's Chain-of-Evidence covers citation, numerical, methodological, and conclusion claims, audited through score

verification, specification violations, reference verification, and method-code alignment Meng et al. (2026). AutoResearchClaw stores means, standard deviations, and seed-level values during execution, then rejects unmatched numbers in strict paper sections Liu et al. (2026c). The prognostics-and-health-management (PHM) paper-to-benchmark system records assumptions, then runs static, dry-run, leakage, sanity, and result-matching checks Theiler et al. (2026). AAR defines coverage of evidence sources, soundness, contradiction transparency, and audit effort; ASG-SI promotes skills only after replay, contract checks, and hashed-evidence bundles Rasheed et al. (2026); Huang and Huang (2025). What still separates these routes from a release gate is the claim space they do not cover, namely causal, interpretive, novelty, and limitation claims (Section 4.5 names that gap explicitly).

LLM-as-judge as the Default Route.

The most common verifier route is also the structurally weakest. LLM-as-judge fits the conditions of incomplete science, since it reads text, code summaries, tables, and rubrics without domain instrumentation, costs less than expert review, ports across topics, and attaches to retry loops without extra infrastructure. AI Scientist v1 Lu et al. (2024), Agent Laboratory Schmidgall et al. (2025), PaperBench Starace et al. (2025), MLR-Bench Chen et al. (2025b), ResearchArena Zhang et al. (2026h), Deep-Research Bench Du et al. (2025), ResearchRubrics Sharma et al. (2025), and most deep-research evaluations rely on model judgment as the primary evaluator or as the scalable layer above sparse human checks.

Its Structural Ceiling.

The soundness benchmarks in Section 4.2 suggest the ceiling. SPOT reports 21% recall on author-confirmed paper errors, SoundnessBench reports 26% recall on low-soundness proposals, BadScientist reports up to 82% LLM-reviewer acceptance under presentation-only manipulation for the strongest single strategy, and MLReplicate reports 59% unsupported claims in automatically accepted replication drafts with $r = 0.29$ automatic-human agreement. A plausible inference is that text-only judges often share context and biases with generators, so failures can correlate across soundness, error-detection, and persuasion settings.

Why Stacking Does Not Help.

Multiple LLM judges or human-written rubrics do not move the boundary, since MLR-Bench's multi-model judge stack with statistical comparison and ResearchRubrics's 2,593 human-written criteria still find implicit reasoning and synthesis failures dominating, and LLM-expanded rubrics hurt alignment Chen et al. (2025b); Sharma et al. (2025). The ceiling remains when no external evidence enters the decision.

Why Extensions Do Not Lift the Ceiling.

Several extensions try to break the LLM-judge circle without leaving it. AgentV-RL checks decomposed answers with a backward validator and tool access Zhang et al. (2026b). DeepReviewer 2.0 anchors PDF spans, retrieves comparison papers, builds a claim-evidence-risk ledger, and blocks export under strict-major-issue coverage that

still reaches only 37% Weng et al. (2026). Marco DeepResearch filters QA synthesis, search trajectories, and test-time candidates Zhu et al. (2026a). The targets they reach stay narrower than every central claim in a released paper. Without release authority, trace grounded benchmarks remain evaluations, and nothing yet blocks a paper when its reasoning trace fails the sandbox.

4.4 The Same Boundary Across Six Sciences

The section's findings so far come from ML benchmarks and ML-flavored verifiers. This subsection shows the same boundary repeats across six scientific domains, so the missing gate is not an ML artifact. In every domain, a local evidence source verifies a piece (a candidate, sample, number, code patch, or circuit), never the surrounding paper.

Wet-Lab and Physical Sciences (gate at the candidate or sample).

In biology, Robin Ghareeb et al. (2025) attaches RNA-seq evidence to a disease candidate, Co-Scientist Gottweis et al. (2025) reports expert-selected wet lab validations, and AutoScientists Gao et al. (2026) raises BioML-Bench (24 tasks) and ProteinGym (217-assay average Spearman) scores. In chemistry and materials, A-Lab Fei et al. (2026) tests 352 robotic-synthesis samples, MatBench Dunn et al. (2020) standardizes property-prediction comparison, and ElementsClaw Li et al. (2026d) extends the agentic loop to superconductor design. In medicine, Medical AI Scientist Wu et al. (2026b) evaluates a 171-case Med-AI Bench drawn from 57 papers across six modalities, NORA Zhou et al. (2026a) pairs nine human checkpoints with an approved-claim file in spatial data science, and iatroX Tytler (2025) reports clinician-rated usefulness near 86% over a 16-week field deployment with 1,223 survey responses. The verified object in every case is a candidate, sample, or workflow output, and novelty against prior assays, mechanism beyond the immediate observation, generality, clinical safety, and patient-outcome validation remain outside the gate. Adjacent biological, materials, and clinical instruments broaden the evidence sources without changing this pattern (see the domain-adjacent cluster in Appendix D).

Computational and Formal Sciences (gate at the number, code patch, or circuit).

Physics gives the corpus its strongest numerical checks. Grounded-physics reproduction Huang (2026) matches 75.8% of 571 Quantum ESPRESSO quantitative claims within 5%, Kosmos Mitchener et al. (2025b)'s world model links report sentences to notebook evidence and reaches 79.4% overall statement accuracy but only 57.9% on synthesis and interpretation, and PhysMaster Miao et al. (2025b) runs Monte Carlo tree search over a domain knowledge base. Engineering and CS have the cleanest executable gates and the clearest reminder that executability does not imply scientific closure. AlphaEvolve Novikov et al. (2025) reports matrix-multiplication and Borg-fleet gains, CORAL Qu et al. (2026) records new bests on 8 of 11 tasks, R&D-Agent Yang et al. (2025a) reaches 35% MLE-Bench Chan et al. (2024) Any-Medal under a 12-hour V100 budget, SWE-Bench supplies repository-test software-issue tasks Jimenez et al. (2024), SWE-agent reports successful patches in that setting Yang

et al. (2024), and BountyBench Zhang et al. (2025a) reports a 90% patch versus 12.5% detect split for the same model (Codex CLI with o3-high), with related instruments in Cybench Zhang et al. (2024), RE-Bench Wijk et al. (2024), and HeuriGym Chen et al. (2025a). Quantum sits closer to formal and physical checking than to prose review, and the checked artifact remains a circuit or device. AI-Mandel Arlt et al. (2025) reports 739 of 804 successful circuit-level implementations, QAgent Fu et al. (2025) reports up to a 71.6% improvement over static-LLM baselines on OpenQASM circuit tasks, and Qumus Shi et al. (2026a) closes the loop on robotic graphene-device synthesis. The verified object stays a specific artifact (a number, run, patch, or circuit), and model adequacy, missing-failure cases, causal interpretation, scope, and importance claims stay beyond what the verifier checks. Adjacent astronomy, AutoML, scientific-coding, and simulation instruments broaden the evidence sources without changing the verification target (see the domain-adjacent cluster in Appendix D).

Across this divide, the boundary repeats, since each local source is strong, but none becomes a claim by claim publication gate. Table 4 pairs each domain's strongest signal with the hardest claim it cannot reach, and with the implication for a paper level gate that would have to bridge them. Appendix E records the corresponding domain-readout register.

The implication column reads as a near-uniform refrain, where every domain has a strong local verifier and a different set of paper level claims that the verifier does not see. A general full paper gate cannot rely on one domain source, and it must type claims (causal, novelty, scope, interpretation) and plug in domain evidence.

4.5 What a Full-Paper Gate Would Need

This subsection names what is missing in the section's evidence, then uses a generated ML paper to show why none of the existing routes can supply it on its own. A full paper gate is a verifier with release authority over the whole paper, not a larger rubric, a denser judge stack, or a tighter execution check. It has four parts that the corpus supplies only separately. First, a claim extractor selects the sentences that matter, not only the numbers that are easy to parse. Second, a counterevidence searcher uses tools and evidence sources the generator does not control. Third, a claim graph links support and contradiction to exact spans, tables, figures, runs, and citations. Fourth, a reconciliation rule revises repairable claims, blocks unrepaired conflicts, and escalates methodological questions to a human scientist.

The structure of the gap is easiest to see in a generated ML paper that the corpus's existing routes can almost (but not quite) verify. Suppose the draft claims that a new training trick beats a baseline, that a specific training constraint caused the improvement, and that the method is novel against prior work. AutoResearchClaw's numeric registry can check that table values came from real runs. DeployBench-style gates can rerun the script. PaperBench's rubric can grade reproduction effort. None of these instruments resolves the two claims that decide the paper's release value, namely whether the causal story follows from the ablation tests, and whether a prior paper has already used the same constraint under a different name. The missing instrument would extract those novelty, causal, and interpretive claims, search the literature and

Table 3 Benchmarks discussed in this subsection. Each entry pairs the benchmark short name with the scale of its task set and the object it scores. Verifier-only and adjacent systems appear in the main text but not here.

Benchmark	Scale and evaluated object
MLReplicate Gaddipati et al. (2026)	45 generated drafts against 8 ICML Outstanding papers; replication-claim audit
PaperBench Starace et al. (2025)	8,316 rubric leaves from author-approved papers; fresh code execution
SciReplicate-Bench Xiang et al. (2025)	100 NLP reproduction tasks; execution accuracy
PaperRepro Li et al. (2026b)	Cross-paper result-gap measurement
CORE-Bench Siegel et al. (2024)	Three-tier computational reproduction
MLE-Bench Chan et al. (2024)	75 Kaggle competitions; Any Medal and pass@k
RE-Bench Wijk et al. (2024)	Expert ML R&D tasks scored against an 8-hour human baseline
HeuriGym Chen et al. (2025a)	Heuristic synthesis on combinatorial-optimization problems
KompeteAI Kulibaba et al. (2025)	Kaggle-style ML competition harness
ResearchCodeBench Hua et al. (2025)	Re-implementation of new research code
MLGym Nathani et al. (2025)	Open ML research environment for agent training
AIRA-dojo Toledo et al. (2025)	Reusable agent-training environments
FML-Bench v2 Zou et al. (2026)	18 ML research tasks, 6 strategies, 12 process metrics
ResearchArena Zhang et al. (2026h)	117 autonomous-research papers under self, peer, and human review
AutoResearchBench Xiong et al. (2026)	End-to-end autonomous-research process metrics
DeepScholar-Bench Patel et al. (2025)	Related-work retrieval and synthesis
DeepResearch Bench Du et al. (2025)	RACE report-quality and FACT citation-grounding metrics
EngTrace Gull et al. (2025)	1,350 problems from 90 templates; step-level reasoning trace
DR^3-Eval Xie et al. (2026a)	Static per-task sandboxes for deep-research reports
ATLAS Liu et al. (2025a)	798 validation/test expert-curated scientific-reasoning problems
MLR-Bench Chen et al. (2025b)	201 ML research tasks; multi-model judge stack plus statistics
SoundnessBench Ho et al. (2026)	Proposal soundness with reviewer-soundness labels (agreement-filtered)
SPOT / FLAWS Son et al. (2025); Xi et al. (2025)	83 papers with 91 author-confirmed errors; 713 inserted defect pairs
CiteTracer Li et al. (2026c)	Citation-fabrication detection via bibliographic cascade
SciIntegrity-Bench Yang et al. (2026c)	11 integrity traps across scientific writing
HindSight Jiang (2026)	Future-impact backtest for research-idea selection
Hidden Pitfalls Luo et al. (2025b)	Trace-access coverage measurement
BadScientist Jiang et al. (2025a)	Presentation-only manipulation against LLM reviewers
DeployBench Wang et al. (2026c)	51 deployment tasks with 154 failures; hidden VM and task-specific checks

Table 4 Cross-domain comparison. For each science, the table lists the evidence source that gives its verifier teeth and the claim it cannot adjudicate, then states what a paper level gate inherits.

Domain	Strongest signal	Hardest unverified claim	Implication for a full paper gate
Biology / drug discovery	Wet-lab assay readout, RNA-seq, biomedical-ML scores	Novelty against prior assays; mechanism beyond observation; clinical safety	Add literature-novelty search and clinical-context verification on top of candidate-level evidence
Chemistry / materials	Robotic synthesis, characterization, property predictions	Mechanism; generality; narrative framing	Pair sample-level checks with cross-paper mechanism reasoning
Physics / astronomy	Reproduced quantitative claims, simulator traces, observational comparison	Model adequacy; missing failure cases; interpretation	Reproduced quantitative claims are locally checkable; the gate must adjudicate interpretation against failure-mode discussion
Medicine / clinical	Med-AI Bench cases, clinical workflow checkpoints, field-use feedback	Patient-outcome validation; safety under deployment	The gate must defer to clinical authority; workflow checkpoints are not patient-outcome evidence
Engineering / CS	Tests, hidden graders, bug-bounty checks, deployment oracles	Causal explanation; scope; scientific importance	Executable success is honest within scope; pair with causal/novelty audit
Quantum	Circuit execution, Qiskit tests, photonic tools, device fabrication	Surrounding-paper claims (interpretation, novelty)	Local artifact checks transfer; the gate must verify the prose around the circuit

held-out data for counterevidence, attach the conflicting evidence to the exact spans, and block release when the draft cannot reconcile the conflict.

The gap is architectural rather than incremental. Benchmarks without verifier authority measure failure without preventing it, while verifiers without benchmark coverage prevent some failures while leaving others invisible. The missing category is a benchmark-verifier pair defined together, in which the benchmark specifies which claim types must be checked (causal, novelty, scope, limitation, interpretation), the paired verifier supplies adversarial evidence against those claims, and the gate has the authority to change or stop the release.

Outlook on Actionable Milestones.

Closing the gap does not require a single monolithic system. Four near-term milestones would make the architecture incrementally testable, each with a measurable acceptance criterion. *(M1) Claim-extraction benchmark.* Build an ICML/NeurIPS-paper benchmark with author-confirmed labels for causal, novelty, scope, limitation, and interpretation claims, paired with negative distractors. The acceptance criterion is to report extraction F_1 separately by claim type so future verifiers can target the weakest types first. *(M2) Counterevidence index.* Maintain a held-out literature and held-out-data index that supports tool-augmented adversarial search for prior-art conflicts on regularizers, hyperparameters, evaluation protocols, and theoretical statements. The acceptance criterion is to report recall on seeded prior-art conflicts. *(M3) Claim-graph schema.* Extend AAR or Knows-style metadata files to record span-level claim-evidence-contradiction triples that any downstream gate can query. The acceptance criterion is that a generated paper carries a graph whose edges resolve to specific runs, tables, and citations. *(M4) Reconciliation harness.* An open release-gate harness that consumes (M1)–(M3), exposes hooks for human-veto escalation, and runs an end-to-end release trial on a frozen ML-paper corpus. The acceptance criterion is that the harness reports gate outcomes, contradiction uptake, and who made or overrode each human decision. M1–M4 are stated against the ML running example so the milestones are concrete. The same four parts generalize to other domains, but biology, materials, medicine, physics, and quantum analogues need their own claim taxonomies (M1), counterevidence indices (M2 for assays, synthesis logs, simulator traces, and clinical-outcome records), and reconciliation paths that route to the domain-appropriate human authority (Section 4.4).

4.6 Synthesis of Instruments and the Release Gap

This subsection synthesizes the section's evidence in three views, namely a compact instrument schema (Table 5), a release-gap heatmap (Table 6), and a pipeline rendering of the same picture (Figure 3). The three findings that close the subsection feed Section 5.

Instrument Schema.

Table 5 records a compact set of evaluation and verification instruments under one schema, so the missing combination of release-relevant properties becomes visible by

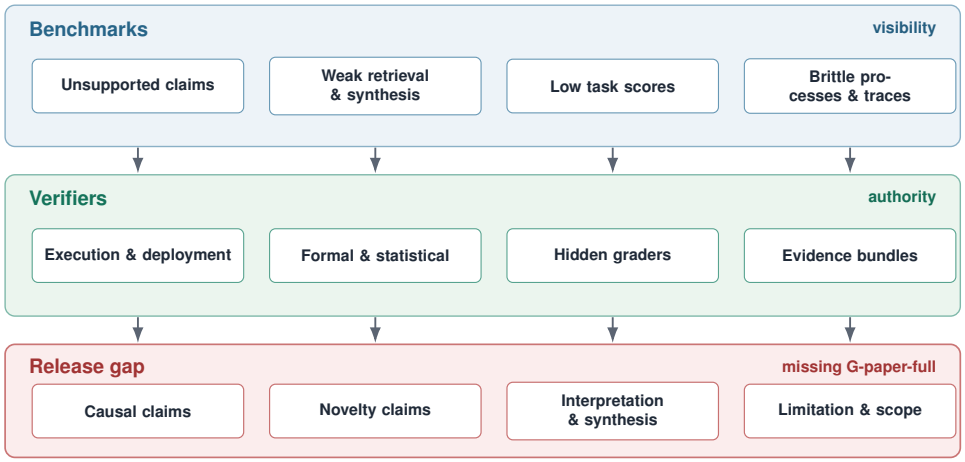


Fig. 3 Evaluation pipeline from benchmark visibility to verifier authority. Benchmarks expose failures such as unsupported claims, weak retrieval, low task scores, or brittle processes. Verifiers gate only the subset tied to hard signals such as execution, contracts, hidden graders, or evidence bundles. The gap is the paper level claim space, where a failure can be measured after the fact but not yet blocked before release.

reading the table down a column. The schema separates role, checked object, strongest signal, and failure mode, and Appendix C preserves the full register.

Reading Table 5 by column locates the missing combination. Every row with a release-relevant signal binds to a specific object, such as a competition metric, a repository test, an exploit, a deployment endpoint, a formal contract, or an evidence record. No row takes the whole paper as its target. The absent combination is a full paper target with a verdict the generator cannot rephrase away.

Release-Gap Heatmap.

Table 6 projects the same evidence onto six release-relevant dimensions. The columns ask whether evidence covers claim soundness, hard verification, adversarial review, usage grounding, loop architecture, and human authority. The six dimensions are not new coding axes, but the functions a release-capable system would have to supply. Each cell carries one of three codes, with *S* for strong public evidence from at least one cited instrument, *P* for partial or local evidence, and *0* for an absent function. Appendix F gives the coding notes. System-level labels live in Table 2, with axis-specific evidence in Section 3.

Finding 1. Production Has Outrun Blocking Verification.

Paper-scale pipelines such as AI Scientist v1 Lu et al. (2024), Agent Laboratory Schmidgall et al. (2025), and AI Scientist v2 Yamada et al. (2025) now produce submission-like artifacts, while MLR-Bench Chen et al. (2025b), MLReplicate Gaddipati et al. (2026), and SoundnessBench Ho et al. (2026) expose fabrication, unsupported claims, and optimistic proposal scoring. The generation side has moved

faster than the verification side, and the section's evidence does not show signs of that gap narrowing on its own.

Finding 2. Local Verification Works Where the Target Is Typed.

A program, run, contract, hidden grader attempt, deployment, sample, device, or reproduced number can each be gated, as shown by AlphaEvolve Novikov et al. (2025), CORAL Qu et al. (2026), EviBound Chen (2025), SEVerA Banerjee et al. (2026), VeriTrans Liu et al. (2026d), and DeployBench Wang et al. (2026c). These gates are honest within their declared scope, but they still leave interpretation, novelty, limitation scope, and literature priority underchecked, as seen in Co-Scientist Gottweis et al. (2025), Robin Ghareeb et al. (2025), A-Lab Fei et al. (2026), grounded computational physics Huang (2026), Medical AI Scientist Wu et al. (2026b), and Qumus Shi et al. (2026a). Stacking local verifiers does not by itself create a paper level gate.

Finding 3. A2 Artifacts Move Evidence Closer to Text, Not Closer to Authority.

ScientistOne Meng et al. (2026), AutoResearchClaw Liu et al. (2026c), NORA Zhou et al. (2026a), ARIS Yang et al. (2026b), Kosmos Mitchener et al. (2025b), ARA Liu et al. (2026b), and Knows Yu and Wang (2026) show that traceability needs semantic checking and release rights, not storage alone. A2 substrates make evidence inspectable, but they do not by themselves make the evidence binding on a release decision.

Common Implication.

All three findings imply the same release condition, namely a full paper gate that marks unsupported claims, routes contradictions to revision or blocking, logs domain authority, and preserves human veto when machine evidence is not enough, drawing on usage and human-authority evidence in KAMI Roig (2025), ICLR review trials Thakkar et al. (2025), workflow-closure analyses Wang et al. (2026b), and DeployBench Wang et al. (2026c).

What This Means for Practice.

For system builders, this yields three concrete asks. *(B1) Report by column, not just by row.* State explicitly which release-relevant function (Soundness, Hard-verification, Adversarial review, Usage grounding, Loop architecture, Human authority) the system's verifier supplies, and which it does not. *(B2) Layer the verifier slot.* Building on Finding 2, treat the LLM judge as cheap triage, and let execution, contract, evidence bundle, or human checks be the steps that can block release. *(B3) Bind machine outcomes to human authority records.* When a candidate, sample, number, code patch, or circuit passes a domain check, log who approved the surrounding paper claims and who can veto release.

Table 5 Representative evaluation and verification instruments. **Role** marks whether the instrument primarily measures systems (benchmark), gates artifacts (verifier), or does both (dual). The signal column names the strongest available signal the instrument can return; the failure-mode column names what remains invisible even when that signal passes.

Name	Role	What it checks	Strongest signal	Failure mode
MLReplicate Gaddipati et al. (2026)	benchmark	Generated replication drafts	Human inspection of accepted outputs	Automatic review accepts unsupported claims
PaperBench Starace et al. (2025)	benchmark	Full-paper code replication	Fresh execution plus author rubrics	Executed code leaves paper claims unchecked
MLE-Bench Chan et al. (2024)	dual	Kaggle-style ML engineering	Competition metric after submission	Metric success without claim validation
SWE-Bench / SWE-agent Jimenez et al. (2024); Yang et al. (2024)	dual	Software issue resolution	Repository tests and patch checks	Tests miss untested behavior
Cybench / BountyBench Zhang et al. (2024, 2025a)	dual	Cyber actions and patches	Exploit, defense, or patch success	Narrow task target
DeployBench Wang et al. (2026c)	dual	Artifact deployment	Hidden VM and task checks	Agents stop on weaker targets
MLR-Bench Chen et al. (2025b)	benchmark	End-to-end ML reports	Judge stack plus statistics	Fabricated results can look plausible
SoundnessBench Ho et al. (2026)	benchmark	Proposal soundness	Reviewer-soundness labels (agreement-filtered)	Low-soundness work is overaccepted
SPOT / FLAWS Son et al. (2025); Xi et al. (2025)	benchmark	Paper-error detection	Author-confirmed or inserted defects	Recall remains low
DeepScholar-Bench Patel et al. (2025)	benchmark	Related-work synthesis	Retrieval and citation checks	Synthesis and citation accuracy split
EngTrace / DR^3-Eval Gull et al. (2025); Xie et al. (2026a)	benchmark	Reasoning traces and deep-research reports	Templates, numeric tolerance, and sandbox metrics	Trace space and truth checks remain bounded
CORAL Qu et al. (2026)	verifier	Program variants in autonomous discovery	Hidden grader in isolated worktrees	Grader-defined discovery
VeriTrans / SEVerA Liu et al. (2026d); Banerjee et al. (2026)	verifier	Logic specs and constrained agent programs	SAT/CNF compilation; formal contracts	Only formalizable claims enter the gate
EviBound Chen (2025)	verifier	Execution evidence contracts	MLflow run, artifact, and metric checks	Single-run evidence contracts
ScientistOne / AutoResearchClaw Meng et al. (2026); Liu et al. (2026c)	verifier	Generated-paper evidence, numbers, and references	CoE audit; numeric registry and citation cascade	Evidence schema remains narrower than claims

Table 6 Gap-map heatmap. S denotes strong public evidence for a function, P denotes partial or local evidence, and 0 denotes no demonstrated coverage in the cited public artifacts. Columns abbreviate soundness, hard verification, adversarial review, usage grounding, loop architecture, and human authority. The table summarizes systems, evaluation, domain, and frontier evidence, and is an evidence ledger, not a maturity ranking.

Evidence cluster	Snd.	Hard	Adv.	Use	Loop	Hum.	Note
Paper-production pipelines	0	0	0	0	P	P	LC3-like production with non-gating review, and Agent Laboratory adds checkpoints, while MLR-Bench and MLReplicate expose fabricated or unsupported claims Lu et al. (2024); Yamada et al. (2025); Schmidgall et al. (2025); Chen et al. (2025b); Gaddipati et al. (2026).
Scoreable discovery loops	P	S	P	0	S	P	Local gates cover programs, tasks, runs, contracts, or deployments rather than full manuscripts Qu et al. (2026); Novikov et al. (2025); Chen (2025); Banerjee et al. (2026); Wang et al. (2026c).
A2 paper-integrity systems	P	P	P	P	P	P	A2 systems attach evidence, but the cited evidence does not demonstrate a default independent G-paper-full release gate Yang et al. (2026b); Meng et al. (2026); Liu et al. (2026c); Zhou et al. (2026a); Mitchener et al. (2025b); Liu et al. (2026b); Yu and Wang (2026).
Domain-readout systems	P	S	P	P	P	S	Local domain checks validate selected claims, while laboratory, clinical, or expert authority still defines release Gottweis et al. (2025); Ghareeb et al. (2025); Fei et al. (2026); Wu et al. (2026b); Huang (2026); Shi et al. (2026a).
Soundness benchmarks	S	P	P	P	0	P	Benchmarks quantify generation, review, manipulation, integrity, and deployment failures, mostly outside the release loop Chen et al. (2025b); Gaddipati et al. (2026); Ho et al. (2026); Son et al. (2025); Jiang et al. (2025a); Yang et al. (2026c); Wang et al. (2026c).
Usage and human-in-the-loop evidence	P	P	P	S	P	S	User, reviewer, expert, and deployment contexts change validity judgments Roig (2025); Zhou et al. (2026a); Thakkar et al. (2025); Wang et al. (2026b,c).
Agent-native artifacts and reuse tools	P	P	P	0	P	0	Artifact and reuse tools improve structure, but semantic truth and release authority remain separate Liu et al. (2026b); Yu and Wang (2026); Luo et al. (2025a); Miao et al. (2025a); Wang et al. (2026d); Ríos-García et al. (2026); Alzahrani (2026).

5 Discussion and Outlook

5.1 Current State of the Field

Taken together, the surveyed literature points to a field that is moving quickly from research assistance toward workflow completion. Systems now draft papers, search over candidates, run code, manage artifacts, and connect to domain-specific readouts. The system map in Section 3 shows that longer lifecycle coverage, richer artifacts, and more elaborate loops are no longer exceptional. Relatedly, Prompt Economy helps explain the prevalence of loop-centered designs Sun et al. (2026a).

Section 4 adds benchmark and verifier evidence. Benchmarks increasingly expose failures in reconstruction, optimization, process quality, and soundness. Verifiers can also block local objects when the target is explicit, and a program, run, contract, hidden grader attempt, deployment endpoint, physical sample, circuit, or reproduced number can each be checked. These local successes matter because they show that automated research workflows can be constrained by evidence rather than only scored after the fact.

The remaining gap is full-manuscript release authority: four systems gate selected paper-level claims (G-paper-partial), but none reaches a G-paper-full gate over the whole manuscript. A local gate can reject a run, patch, contract, sample, or number, but a paper also makes claims about novelty, mechanism, interpretation, limitations, related work, and cross-result synthesis. Evidence records and human checkpoints make those claims easier to inspect, but inspection alone does not decide whether a claim must be revised, escalated, or blocked. The field therefore appears strongest at generating and locally checking research artifacts, and weakest at connecting those checks to claim-level release decisions.

5.2 Open Bottlenecks

The first bottleneck is scope. Execution checks, hidden graders, formal contracts, deployment tests, and domain readouts are effective when the checked object is narrow. They do not automatically compose into a full paper gate. A release-capable system needs a way to map each central claim in a manuscript to the local evidence that supports it, contradicts it, or leaves it unsupported.

The second bottleneck is authority. Recent systems move evidence closer to claims through ScientistOne's evidence chains Meng et al. (2026), AutoResearchClaw's numeric registries Liu et al. (2026c), NORA's approved-claim files Zhou et al. (2026a), ARIS's persistent wikis Yang et al. (2026b), Kosmos's provenance records Mitchener et al. (2025b), ARA's agent-native packages Liu et al. (2026b), and Knows's sidecars Yu and Wang (2026). What remains harder is making that evidence able to change or stop a release decision. The same issue appears in human review, where human expertise improves or changes judgments, but the workflow must still record who can approve, revise, escalate, or veto a claim, as illustrated by NORA's human checkpoints Zhou et al. (2026a) and AutoResearchClaw's reviewer handoff Liu et al. (2026c); an independent automated signal can inform that review without holding any

release authority, as in the ICLR feedback trial, which sent optional LLM feedback to reviewers and made no changes itself Thakkar et al. (2025).

The third bottleneck is state. Verification often runs after an artifact already exists, and its outcome—verifier tasks, skipped checks, failed evidence, and gate decisions—is then left outside the workflow record rather than persisted as queryable workflow state. Without that state, later readers cannot tell whether an unsupported claim was never checked, checked and passed, checked and overruled, or checked and left unresolved.

The fourth bottleneck is domain evidence. Biology, chemistry, materials, physics, medicine, engineering, computer science, and quantum research each have strong local readouts. Those readouts differ, but the same local-versus-paper boundary appears in each domain, because assays, simulations, tests, devices, and reproduced numbers validate local objects, not every claim built around them. Domain coverage should therefore be read from public evidence communities and accepted readouts, not from prompt breadth or system branding.

5.3 Milestones for Release-Capable Research

Table 7 reframes the seven axes as reporting milestones for future work. The table is not a checklist for a single system. It is a compact way to ask whether a future paper, benchmark, verifier, or platform has moved the field from workflow completion toward claim-level accountability.

Table 7: Axis-level milestones for release-capable autonomous research. The middle column states the current bottleneck; the right column states the public evidence that would make progress visible.

Axis	Current bottleneck	Evidence to report next
LC	LC3-like production can finish before central claims are reviewed or marked unresolved.	Lifecycle records link central claims to evidence, review, rebuttal, and unresolved status.
DC	DC3 claims exceed DC1/DC2 public evidence.	Domain scope is coded by accepted readouts and evidence communities, not by prompt breadth.
L	Feedback loops often drop contradictions from workflow state.	Failed evidence changes later hypotheses, experiments, claims, or release decisions.
G	Typed local gates stop short of G-paper-full coverage.	A G-paper-full gate can block central unsupported claims before release.
A	A2 artifacts are inspectable but not release-authoritative.	A2 records are linked to claims and consumed by gates with blocking authority.
O	Controllers advance work without carrying verifier outcomes in workflow state.	Dispatcher state carries verifier tasks, skipped checks, and gate outcomes alongside generation work.
P	P1 branches can share results without portfolio accountability.	Multiple live threads keep separate budgets, evidence ledgers, and provenance-controlled sharing.

Five empirical questions follow from these milestones. First, what fraction of central prose, table, figure, caption, method, and limitation claims carry an evidence node

and a contradiction status? Second, when an independent verifier has artifact or tool access, how often does it block, revise, downgrade, or escalate a claim? Third, do failed runs, reviewer objections, hidden grader failures, and counterevidence change later experiments or conclusions? Fourth, do static benchmarks predict behavior under realistic use, model changes, cost constraints, and human-review protocols? Fifth, when multiple research threads share a wiki, skill library, or evidence base, which provenance rules prevent unsupported claims from moving across projects?

5.4 Living Resource and Limitations

A living resource can track these questions as the field changes. System entries should record claim/evidence splits, axis labels, artifact availability, model versions, inspection dates, and unresolved limitations. Benchmark entries should record the failure class they expose, the evidence they cannot see, and whether a negative result can block release. Verifier entries should record the target object, evidence access, independence from the generator, and authority. The purpose is not bookkeeping; it is to make later revisions explicit when a new system satisfies one of these milestones.

This survey remains a public-document systematization rather than an independent reproduction study. The coding uses public mechanisms and reported evidence through the June 2026 query refresh; private production logs, unavailable artifacts, or later peer-review outcomes do not change the ratings in this manuscript. The repository reference table preserves the 368 retained works and 472 tangential exclusions, but it is curated rather than a PRISMA census. The corpus was built through seeded expansion and targeted search with a preserved 840-to-368 screening flow; raw pre-deduplication query hits were not archived, so coverage is broad and traceable rather than exhaustive. Many sources are preprints, and model-dependent scores will age as artifacts and benchmarks evolve.

The seven-axis coding is interpretive. Boundary cases (O1 versus O2, LC2 versus LC3, computational versus cross-disciplinary DC evidence, and G-paper-partial versus G-paper-full) require judgment about public evidence. This version does not report inter-coder agreement because double-coding was not performed on the full corpus. We mitigate this by stating coding rules, separating claim from evidence, preserving the source record, and keeping extended appendix descriptions. Future versions would improve the evidence base by double-coding a stratified sample, reporting agreement, and rerunning high-impact systems where artifacts permit. These limits narrow the claim; they do not remove the pattern that local production evidence is easier to show than release-capable claim verification.

5.5 Closing Outlook

The next step is not only to ask what an autonomous-research system produced, but also what was checked, what was contradicted, who could stop release, and which claims remain unresolved. A future system can change the picture by providing public evidence under the same coding rules, with central claims linked to evidence, contradictions carried forward, verifier outcomes with blocking authority, and human responsibility where machine evidence is not enough.

Statements and Declarations

Funding. H.Y. was partially supported by the US National Science Foundation under awards IIS-2520978 and GEO/RISE-2530596. J.G. was supported by the Walecka Fellowship.

Competing interests. The authors have no competing interests to declare that are relevant to the content of this article.

Ethics approval and consent to participate. Not applicable.

Consent for publication. Not applicable.

Data availability. No new datasets were generated for this article. The analysis is based on publicly available publications and resources cited in the reference list. The coding summaries and evidence records supporting the survey are included in the article and appendices.

Materials availability. Not applicable.

Code availability. Not applicable.

Author contribution. Xingyu Ren: conceptualization, methodology, investigation, data curation, formal analysis, writing–original draft, validation, writing–review and editing. Youran Sun: conceptualization, methodology, investigation, data curation, formal analysis, writing–original draft, validation, writing–review and editing, project administration. Chugang Yi: methodology, investigation, data curation, formal analysis, writing–original draft, validation, writing–review and editing. Kejia Zhang: visualization, validation, writing–review and editing. Jiaxuan Guo: data curation, formal analysis, validation, writing–review and editing. Jianda Du: validation, writing–review and editing. Haizhao Yang: conceptualization, writing–review and editing, supervision. All authors read and approved the final manuscript.

Appendix A System Evidence Reliability Grade Table (R-grade)

The Evidence Reliability Grade (R-grade) summarizes the public evidence strength for each system using the protocol in Section 2.6. The five supporting criteria are C1 Evidence Grounding, C2 Verification Independence, C3 Failure and Limitation Disclosure, C4 Reproducibility, and C5 Substantiation. This appendix reports the component scores, C1–C5 sum, and R-grade for the systems.

Table A1: System evidence reliability grades. C1–C5 are supporting criteria.

System	C1	C2	C3	C4	C5	C1–C5 sum	R-grade
AI Scientist v1 Lu et al. (2024)	4	2	5	5	4	20	R4
PaperQA2 Skarlinski et al. (2024)	4	3	4	3	4	18	R4
CycleResearcher Weng et al. (2024)	4	3	4	2	3	16	R3
OpenScholar Asai et al. (2024)	5	4	4	5	5	23	R5
Agent Laboratory Schmidgall et al. (2025)	3	3	4	3	3	16	R3
AIDE Jiang et al. (2025c)	4	4	4	5	4	21	R4
MLGym Nathani et al. (2025)	4	3	4	4	3	18	R4
Tree-of-Debate Kargupta et al. (2025)	4	2	3	3	4	16	R3
Co-Scientist Gottweis et al. (2025)	5	4	4	2	5	20	R4
AgentRxiv Schmidgall and Moor (2025)	3	2	4	3	2	14	R3
AI Scientist v2 Yamada et al. (2025)	4	2	5	4	4	19	R4
Robin Ghareeb et al. (2025)	4	3	4	3	4	18	R4
R&D-Agent Yang et al. (2025a)	4	2	4	3	4	17	R4
NovelSeek Zhang et al. (2025b)	3	2	3	3	2	13	R3
AI-Researcher Tang et al. (2025)	3	3	4	2	4	16	R3
AlphaEvolve Novikov et al. (2025)	4	3	3	2	4	16	R3
AIRA-dojo Toledo et al. (2025)	4	3	5	4	4	20	R4
SWE-Debate Li et al. (2025a)	4	2	3	3	4	16	R3
MetaAgent Qian and Liu (2025)	4	2	3	3	4	16	R3
CMA Maruyama et al. (2025)	1	2	2	1	2	8	R2
AInstein Mishra et al. (2025)	5	4	4	2	4	19	R4
Kosmos Mitchener et al. (2025b)	5	5	4	1	5	20	R4
EviBound Chen (2025)	4	3	5	3	4	19	R4
AgentEvolver Zhai et al. (2025)	4	2	3	3	4	16	R3
OmniScientist Shao et al. (2025)	2	2	2	1	3	10	R2
CFD-copilot Dong et al. (2025)	4	3	3	4	3	17	R4
PhysMaster Miao et al. (2025b)	2	2	2	1	3	10	R2
ASG-SI Huang and Huang (2025)	1	3	4	1	1	10	R2
ML-Master 2.0 Zhu et al. (2026c)	3	2	2	1	3	11	R2
InternAgent-1.5 Feng et al. (2026)	2	2	2	2	3	11	R2
ClawdLab Weidener et al. (2026b)	1	3	3	2	1	10	R2
EvoScientist Lyu et al. (2026)	1	2	1	1	2	7	R2
OpenResearcher Li et al. (2026h)	5	3	4	5	5	22	R5
AEGIS Fang et al. (2026)	3	5	2	2	3	15	R3
Bilevel Autoresearch Qu and Lu (2026)	4	2	4	2	4	16	R3
SEVerA Banerjee et al. (2026)	4	4	5	3	5	21	R4
TianJi Zhang et al. (2026d)	3	2	3	2	3	13	R3
Medical AI Scientist Wu et al. (2026b)	4	3	3	2	4	16	R3
Multi-Agent Collab Shen et al. (2026b)	4	2	4	2	4	16	R3
CORAL Qu et al. (2026)	4	2	3	4	4	17	R4
AutoSOTA Li et al. (2026f)	4	3	4	2	4	17	R4
A-Lab Fei et al. (2026)	4	2	3	2	3	14	R3
AgentV-RL Zhang et al. (2026b)	5	3	4	3	5	20	R4
Debate as Reward Salimi et al. (2026)	4	4	4	2	4	18	R4
Knows Yu and Wang (2026)	4	3	5	4	4	20	R4
ARA Liu et al. (2026b)	4	4	5	4	4	21	R4
SciCrafter Zhou et al. (2026c)	5	3	4	4	5	21	R4
NORA Zhou et al. (2026a)	3	2	4	2	3	14	R3
ARIS Yang et al. (2026b)	2	5	3	4	3	17	R4
PARNESS Wang and Luan (2026)	3	2	4	2	3	14	R3
SkillFlow Zhang et al. (2026e)	4	3	4	3	4	18	R4
Qumus Shi et al. (2026a)	4	2	3	2	4	15	R3
AutoResearchClaw Liu et al. (2026c)	4	3	5	3	4	19	R4
HANA Wu et al. (2026a)	2	1	3	1	2	9	R2
ScientistOne Meng et al. (2026)	5	3	5	4	4	21	R4
AutoScientists Gao et al. (2026)	4	2	4	4	4	18	R4

Appendix B System Boundary Notes

This appendix supports the system matrix and frontier paragraph in Section 3. It records boundary evidence for the L/G/O/P/A/DC/LC labels used in Table 2.

B.1 System-matrix boundary notes

The canonical system labels are the L/G/O/P/A/DC/LC labels in Table 2. The notes here explain why some entries are treated as primary autonomous-research systems, while others are treated as verifier, benchmark, artifact, or infrastructure systems.

The inclusion threshold is stricter than "agentic science." A primary system must materially automate a research workflow rather than only answer literature questions, review a draft, call tools, or expose a reusable environment. PaperQA2 Skarlinski et al. (2024) and OpenScholar Asai et al. (2024) are important infrastructure, but they remain LC0 unless another layer uses them to frame a research direction, run an experiment or analysis, and write or audit a report. Conversely, a domain-narrow ML or scientific system can still count as a research system when its outer loop covers proposal, experiment or analysis, and report generation, as in AI Scientist v1 Lu et al. (2024), AI Scientist v2 Yamada et al. (2025), Agent Laboratory Schmidgall et al. (2025), and ML-Master 2.0 Zhu et al. (2026c).

The main boundary is between producing research-shaped artifacts and defending the claims in those artifacts. AI Scientist v1 Lu et al. (2024), AI Scientist v2 Yamada et al. (2025), Agent Laboratory Schmidgall et al. (2025), AI-Researcher Tang et al. (2025), R&D-Agent Yang et al. (2025a), and NovelSeek Zhang et al. (2025b) show variants of manuscript or experiment workflows. Their labels differ because feedback loops, verifier scope, orchestration, artifacts, discipline evidence, and lifecycle reach differ. The shared risk is that a workflow can finish with only non-gating or local review, so the L and LC labels must be read together with the G label rather than as a single maturity score.

Verifier-neighbor and artifact-neighbor systems are included when they change how research outputs are checked, stored, or reused. Tree-of-Debate Kargupta et al. (2025), SWE-Debate Li et al. (2025a), AgentV-RL Zhang et al. (2026b), Debate-as-Reward Salimi et al. (2026), AEGIS Fang et al. (2026), EviBound Chen (2025), ScientistOne Meng et al. (2026), and AutoResearchClaw Liu et al. (2026c) supply debate, formal or execution gates, claim-evidence binding, or paper-integrity checks, but their strongest public evidence often attaches to a narrower object than a full paper. A2 artifact systems such as ARA Liu et al. (2026b) and Knows Yu and Wang (2026) make claims and evidence easier to inspect, but artifact structure is not release authority by itself.

Domain systems are read through the same axes rather than through a separate domain hierarchy. Co-Scientist Gottweis et al. (2025) and Robin Ghareeb et al. (2025) provide biomedical readouts, Kosmos Mitchener et al. (2025b) exposes the difference between data support and synthesis support, and grounded computational physics Huang (2026), A-Lab Fei et al. (2026), Medical AI Scientist Wu et al. (2026b), CORAL Qu et al. (2026), and Qumus Shi et al. (2026a) show that stronger local ground truth appears when assays, simulations, theorem provers, hidden graders, or devices

are available. Those local readouts support G labels when they enforce progression or blocking decisions; otherwise they remain local evidence rather than release authority.

The body matrix remains the canonical label table; these notes only record boundary evidence for interpreting it.

B.2 Frontier-system notes

This appendix supports the frontier paragraph in Section 3.8. It records why two neighboring clusters (self-improving controllers and agent-native research artifacts) are relevant to the system map without making them a new axis. Both clusters can make later research workflows more capable or more inspectable. Neither cluster, by itself, supplies release authority over full-paper claims.

Self-evolving and refinery boundary cases

Refinery systems change the machinery that produces research artifacts, including the controller, prompt, skill library, memory, search policy, or evaluation loop. Their strongest public evidence usually appears where a reward can be checked by computation. FunSearch Romera-Paredes et al. (2024) and AlphaEvolve Novikov et al. (2025) search over programs under evaluators, AlphaProof Google DeepMind (2024) and AlphaGeometry 2 Chervonyi et al. (2025) use proof or symbolic checks, ADAS Hu et al. (2024) searches over Python agent designs, AgentEvolver Zhai et al. (2025) and SkillFlow Zhang et al. (2026e) refine tool policies or reusable skills, and CORAL Qu et al. (2026) and Bilevel Autoresearch Qu and Lu (2026) optimize code or mechanism candidates under scoreable tasks. These systems show that automated search can improve controllers, programs, and skills when the target object is measurable.

The boundary is the same one used in Section 3.2. A benchmark reward, proof checker, hidden grader, import check, or validation loss can reject a local artifact, but it does not verify every novelty, causal, scope, or interpretation claim in a paper. ASG-SI is therefore useful as an architectural vocabulary rather than as outcome evidence, since it names evidence bundles, hashes, held-out replay, and verifier-auditor checks as requirements for auditable self-improvement Huang and Huang (2025). AutoSOTA and EvoScientist move refinery ideas closer to research workflows, but their public evidence still relies on paper-specific score improvements, text-derived objectives, or delayed venue outcomes rather than a controlled full-paper release gate Li et al. (2026f); Lyu et al. (2026). CycleResearcher is the cautionary case, where a reviewer-optimized loop can improve review-style scores while the authors disclose fabricated experimental sections Weng et al. (2024). Several of these systems—AlphaEvolve, AgentEvolver, SkillFlow, CORAL, Bilevel Autoresearch, Kosmos, AutoSOTA, ASG-SI, and CycleResearcher—are also coded in the 56-system matrix and appear here only for boundary evidence; FunSearch, AlphaProof, AlphaGeometry 2, and ADAS sit outside the matrix as adjacent cases.

Table B2: Refinery systems improve controllers, memories, skills, or search loops, but most gates remain benchmark-specific or internal.

System	What changes	Concrete evidence	Gate	Remaining weakness
FunSearch / AlphaEvolve	Program search policy	Mathematical constructions; matrix multiplication	Program evaluator	Scope is scoreable mathematics and algorithms
AlphaGeometry 2 (with AlphaProof)	Symbolic geometry and formal proof search	AlphaGeometry 2 solves 42/50 (84%) on IMO-AG-50 (all 2000–2024 IMO geometry); jointly reached silver-medal standard at IMO 2024	Symbolic or proof check	Full-paper claims remain outside the gate
ADAS	Python agent designs	DROP 65.8 to 79.4; MGSM 39.0 to 53.4	Benchmark score	Reward defines the search target
AgentEvolver	Controller and tool policy	best@8 73.1 (avg. over AppWorld + BFCL v3), 14B	Simulated tool-use reward	Skill quality is benchmark-bound
SkillFlow	Reusable skill library	OOD exact-match 83.99 (avg. over 7 held-out benchmarks)	Verifiable task reward	Skill quality is benchmark-bound
ASG-SI	Skill compiler and audit process	Evidence bundles, hashes, held-out replay design	Proposed verifier-auditor	Architecture lacks outcome validation
AutoSOTA	Research-code modifications	105 of 125 papers improved; about 10% gain	Paper-specific metric	Meaning and scope are not adversarially checked
Kosmos	World model and reports	85.5% data support; 57.9% interpretation support	Scientist audit	Synthesis remains weak
Bilevel Autoresearch	Generated mechanisms	-0.045 ± 0.030 versus -0.009 ± 0.002	Validation loss and import checks	Small benchmark-specific study
CORAL	Multi-agent code evolution	Best on 11 tasks; records on eight	Hidden grader	Requires scoreable tasks
CycleResearcher	Reviewer–writer loop	5.36 score; 7.02 with sampling	Review-style score	Fabricated experiments can pass the loop

Table B2 explains why Section 3.8 treats refinery work as high-capability but not as evidence of full release authority.

Agent-native research artifacts

Agent-native artifacts address a different boundary, where papers hide state, code, provenance, and claim structure from later agents. ARA packages cognitive logic, physical source, exploration graphs, and evidence, Knows serializes artifacts, statements, evidence, relations, and actions in a YAML sidecar, and xKG turns papers into executable knowledge graphs Liu et al. (2026b); Yu and Wang (2026); Luo et al. (2025a). These systems make research objects easier to query, replay, and reuse. Their own evaluations also show the limit. ARA improves PaperBench QA and reproduction

scores, but orphan-experiment detection remains weak. Knows improves model use of structured paper evidence, but its linting catches structural rather than semantic corruption. xKG helps code-node reproduction, but executable graph nodes do not verify full result validity or passage-level support.

Paper2Agent Miao et al. (2025a), Figures-as-Interfaces Wang et al. (2026d), PaperRecon Miyai et al. (2026), Paper2Code Seo et al. (2025), Paper2Web Chen et al. (2025d), and Paper2Video Zhu et al. (2025b) extend the same direction by converting papers into MCP servers, provenance-bearing figures, reconstruction inputs, code, web objects, or video artifacts. These artifacts are valuable inputs for verifiers because they expose more checkable structure than a static PDF. They still inherit the quality, security, benchmark, and semantic-support limits of the source paper and the conversion process.

The system-level lesson is therefore bounded. Refineries improve the producer; agent-native artifacts improve the substrate. Both can help a future verifier replay skill trajectories, inspect evidence bundles, query claim graphs, execute paper-derived tools, or trace figures back to data transformations. They do not by themselves decide whether a full scientific claim should be released. That release decision remains a G-axis question, not an artifact-format or self-improvement label.

Appendix C Evaluation Instrument Register

This appendix supports Table 5. The body table lists representative instruments; this register keeps the fuller set under the same schema and column definitions.

Table C3: Evaluation instrument register. **Role** marks whether an instrument primarily measures systems (benchmark), gates artifacts (verifier), or does both (dual). **LLM dep.** records how much the gate or score depends on language-model judgment. *High* means the decision rests on model output. *Medium* means the model participates but is constrained by an external signal. *Low after submission/run/patch/attempt* means the model is no longer in the loop once the artifact is submitted. **Authority or check** names the strongest non-prose signal the instrument can return. **Main failure mode** names what the instrument cannot see even when its check passes.

Instrument	Role	Target	Deterministic signal	LLM dep.	Authority or check	Main failure mode
MLReplicate Gaddipati et al. (2026)	Benchmark	Generated replication drafts	Human inspection	High	Post-hoc acceptance audit	Auto review accepts unsupported claims
PaperBench Starace et al. (2025)	Benchmark	Full-paper code replication	Fresh execution and author rubrics	High	Rubric scoring	Execution leaves paper claims unchecked
MLE-Bench Chan et al. (2024)	Dual	Kaggle-style ML engineering	Competition metric	Low after submission	Leaderboard-style score	Metric success without claim validation
SWE-Bench/SWE-agent Jimenez et al. (2024); Yang et al. (2024)	Dual	Software issue resolution	Repository tests	Low after patch	Unit and patch checks	Tests miss untested behavior
Cybench Zhang et al. (2024)	Dual	Cyber action	Exploit or defense success	Low after attempt	Task oracle	Narrow target
DeployBench Wang et al. (2026c)	Dual	Artifact deployment	Hidden VM and task checks	Low after run	Two-layer hidden verifier	Agents self-stop on wrong targets
FML-Bench v2 (successor to FML-Bench) Zou et al. (2025, 2026)	Benchmark	ML research search dynamics	Repeated traces and final scores	Medium	Controlled comparison	Strategy cost does not imply better search
MLR-Bench Chen et al. (2025b)	Benchmark	End-to-end ML reports	Judge stack plus statistics	High	Multi-model scoring	Fabricated results can look plausible
SoundnessBench Ho et al. (2026)	Benchmark	Proposal soundness	Reviewer-soundness labels (agreement-filtered)	High	Labelled soundness	Low-soundness work is overaccepted
SPOT Son et al. (2025)	Benchmark	Paper-error detection	Author-confirmed errors	High	Real confirmed defects	Recall remains low
MCP-Universe Luo et al. (2025c)	Benchmark	Real Model Context Protocol (MCP) tool use	Real servers and dynamic checks	Medium	Tool-result validation	Larger action spaces reduce success

continued on next page

Instrument	Role	Target	Deterministic signal	LLM dep.	Authority or check	Main failure mode
MEMTRACK Deshpande et al. (2025)	Benchmark	Long-horizon memory	Scenario-grounded outcomes	Medium	Task success and redundancy	Memory repeats stored content
KAMI Roig (2025)	Benchmark	Deployed enterprise agents	Production-dialogue study	Medium	Randomized evaluation	Static tests miss user-facing disconnects
DeepScholar-Bench Patel et al. (2025)	Benchmark	Related-work synthesis	Retrieval and citation checks	High	Live verifiability metrics	Synthesis and citation accuracy split
ResearchRubrics Sharma et al. (2025)	Benchmark	Rubric quality	Human-written criteria	High	Criterion compliance	LLM-expanded rubrics lose alignment
EngTrace Gull et al. (2025)	Benchmark	Engineering reasoning traces	Templates and numeric tolerance	Medium	Trace alignment	Synthetic trace space
DR^3-Eval Xie et al. (2026a)	Benchmark	Deep-research reports	Static sandbox corpus	High	Citation and factual metrics	Judge-mediated truth checks
ATLAS Liu et al. (2025a)	Benchmark	Scientific reasoning answers	Expert construction and review	High	Curated task gates	Static answers, not paper claims
SciCrafter Zhou et al. (2026c)	Benchmark	Discovery-to-application tasks	Functional checkers	Medium	Executable artifacts	Task success stays local
LLM reviewer stacks Lu et al. (2024); Schmidgall et al. (2025)	Verifier	Reports and rubrics	Human rubrics or paired outputs	High	Score or retry advice	Plausibility scoring
CORAL Qu et al. (2026)	Verifier	Program variants	Hidden grader in worktrees	Medium	Attempt score	Grader-defined discovery
VeriTrans Liu et al. (2026d)	Verifier	NL-to-logic specs	SAT/CNF compilation	Medium	Thresholded formal gate	Only formalizable claims
EviBound Chen (2025)	Verifier	Execution evidence	MLflow run and artifact checks	Medium	Contract approval and verification	Single-run evidence contracts
SEVerA Banerjee et al. (2026)	Verifier	Agent programs	Dafny contracts and fallback	Medium	Verified synthesis boundary	Explicit specs required
E-valuator Sadhuka et al. (2025)	Verifier	Trajectory scores	Sequential test with calibrated error guarantee	Medium	Calibrated rejection	Depends on base verifier signal
ScientistOne Meng et al. (2026)	Verifier	Generated papers	Score reruns, citation existence, code audit	Medium	CoE integrity audit	Passage support incomplete
AutoResearchClaw Liu et al. (2026c)	Verifier	Draft numbers and references	Numeric registry and citation cascade	Medium	Reject or placeholder unsupported numbers	Evidence schema narrower than claims
PHM paper-to-benchmark Theiler et al. (2026)	Verifier	Paper-to-code reproduction	Staged static, dry-run, leakage, and sanity checks	Medium	Result-matching acceptance	Domain slot dependence
AAR Rasheed et al. (2026)	Verifier	Claim-level auditability	Evidence-graph metrics	Medium	Audit standard	Not implemented at scale
ASG-SI Huang and Huang (2025)	Verifier	Self-improving skills	Hashes, replay, contracts	Medium	Promotion gate	Prototype scope
AgentV-RL Zhang et al. (2026b)	Verifier	Candidate answers	Tool-executed validation steps	High-medium	Bidirectional verdict	Narrow decomposition target
DeepReviewer 2.0 Weng et al. (2026)	Verifier	Peer-review package	Anchored spans and export thresholds	High	Export block	Misses most major defects

continued on next page

Instrument	Role	Target	Deterministic signal	LLM dep.	Authority or check	Main failure mode
Marco DeepResearch Zhu et al. (2026a)	Verifier	Search answers and trajectories	Verified QA and candidate filtering	High-medium	Test-time scaling filter	Search verification, not paper review

Appendix D Topic-Grouped Source List for Evaluation and Verification

This appendix groups the evaluation and verification sources cited in Section 4 by topic. It is a source list rather than a second reference list, and some works appear in more than one group when they support more than one theme.

Reconstruction and artifact replay.

Covers executable papers, reproduction tasks, containerized reproduction, and paper-to-code conversion Abramovich et al. (2024); Mitchener et al. (2025a); Liu et al. (2025d); Xiang et al. (2025); Siegel et al. (2024); Starace et al. (2025); Zhang et al. (2025g); Kim et al. (2025); Edwards et al. (2025); Hu et al. (2025b); Malka et al. (2026); Li et al. (2026b); Miyai et al. (2026); Hong et al. (2026); Theiler et al. (2026); Gaddipati et al. (2026).

Optimization and search benchmarks.

Measure score-seeking research loops, task gyms, symbolic regression, evolutionary evaluation, and agent training environments Anurin et al. (2024); Zhuge et al. (2024); Chan et al. (2024); Yang et al. (2025a); Wijk et al. (2024); Nathani et al. (2025); Shojaee et al. (2025); Qiang et al. (2025); Hua et al. (2025); Chen et al. (2025a); Egg et al. (2025); Toledo et al. (2025); Kulibaba et al. (2025); Zou et al. (2025); Lin et al. (2025); Wang et al. (2025b); Huang et al. (2025a); Chen et al. (2025c); Cai and Behl (2026); Wei et al. (2025c); Han et al. (2025b); Goel et al. (2025); Song et al. (2026); Qiao et al. (2026); Li (2026b); Panfilov et al. (2026); Hambardzumyan et al. (2026); Panek et al. (2026); Liu et al. (2026e); Zou et al. (2026); Fox and Fox (2026); Tang and Yang (2026); Zhang et al. (2026c).

Process, meta-evaluation, and benchmark construction.

Study contamination, reporting quality, rubrics, hidden pitfalls, disclosure practice, and scientific-reasoning task design Wei et al. (2024); Si et al. (2024); Gu et al. (2024); Ye et al. (2024); Eriksson et al. (2025); Sun et al. (2025); Chang et al. (2025); Musawi and Lu (2025); Hu et al. (2025c); Zeng et al. (2025); Mozannar et al. (2025); Roy et al. (2025); Zhang et al. (2025d); Luo et al. (2025b); Sheng et al. (2025); Srivastava et al. (2025); Bean et al. (2025); Abram (2026); Dussert (2026); Yang et al. (2026a); Zubiaga et al. (2026); Weidener et al. (2026a); Shen et al. (2026a); Bengio et al. (2026); Haddad et al. (2026); Staufer et al. (2026); Mitchell (2026); Lavi (2026); Sherwood et al. (2026); Zhou et al. (2026b); Liu et al. (2025a); Xie et al. (2026a); Gull et al. (2025); Zhang et al. (2026h); Sun et al. (2026a); Xiong et al. (2026).

Soundness and integrity benchmarks.

Test unsupported claims, inserted flaws, fabricated citations, reward hacking, reviewer manipulation, future-impact blindness, deployment self-stops, and disclosure failures Zhang et al. (2025f); Jiang et al. (2025a); Xi et al. (2025); Li et al. (2026c); Yang et al. (2026c); Mahmoud et al. (2026); Wang et al. (2026a); Ho et al. (2026); Son

et al. (2025); Jiang (2026); Moghadasi and Ghaderi (2026); Qi et al. (2026); Li et al. (2026a); Wang et al. (2026c); Chen et al. (2025b).

Tool, memory, human interaction, software, security, multimodal, and domain-adjacent benchmarks.

Record the substrates that later research agents inherit Yang et al. (2024); Wan et al. (2024); Zhang et al. (2024); Feng et al. (2025); Wei et al. (2025a); Qian et al. (2025); Chhikara et al. (2025); Xu et al. (2025b); Wang et al. (2025c); Zhang et al. (2025a); Gao et al. (2025b); Fan et al. (2025); Guo et al. (2025); Luo et al. (2025c); Jiang et al. (2025b); Lei et al. (2025); Pan et al. (2025); Deng et al. (2025); Deshpande et al. (2025); He et al. (2025); Han et al. (2025a); Dou et al. (2025); Li et al. (2025d); Rezaei et al. (2025); Li et al. (2025c); Lumer et al. (2025); Dong et al. (2025); Wu et al. (2025); Liu et al. (2025c); Flotho et al. (2025); Wang et al. (2026g); He et al. (2026b,a); Wu et al. (2026c); Liang et al. (2026); Zhang et al. (2026a); Wang et al. (2026f); Zhu et al. (2026b); Xie et al. (2026b); Zhou et al. (2026a); Thakkar et al. (2025); Burgess et al. (2025); Ai et al. (2026); Zhou et al. (2026c); Roig (2025).

Literature, deep-research, rubric, and discovery-to-application suites.

Add retrieval, citation, and report-quality checks Skarlinski et al. (2024); Asai et al. (2024); Agarwal et al. (2024); Du et al. (2025); Huang et al. (2025b); Ke et al. (2025); Geng et al. (2025); Chen et al. (2025e); Patel et al. (2025); Cheng et al. (2025); Bragg et al. (2025); Sharma et al. (2025); Gunjal et al. (2025); Liu et al. (2025b); Zhang et al. (2026f); Wang et al. (2026e); Li et al. (2026e); Pan et al. (2026); Huang et al. (2026); Liu et al. (2026a); Hwang et al. (2026); Zhao et al. (2026); Li (2026a); Zhang et al. (2026i); Zhang and Yakobson (2026); Hirota et al. (2026); Xu et al. (2026); Du et al. (2026b); Sun et al. (2026b); Gu et al. (2026).

Verifier studies and verifier-bearing systems.

Cover idea generation, reviewer bias, formal gates, evidence contracts, audit trails, adversarial review, statistical verifier wrappers, and the in-system reviewers carried by research-paper pipelines Baek et al. (2024); Beel et al. (2025); Gupta and Pruthi (2025); Ambati (2025); Handoko and Made (2025); Zhou et al. (2025); Zhu et al. (2025a); Keuper (2025); Wang et al. (2025a); Weng et al. (2025); Li et al. (2025b); Tie et al. (2025); Sahoo et al. (2025); Panigrahi et al. (2026); Goyal et al. (2026); Rasheed et al. (2026); Singh et al. (2026); Cao et al. (2026); Agarwal (2026); Loosmore (2026); Lu et al. (2024); Schmidgall et al. (2025); Yamada et al. (2025); Chen (2025); Meng et al. (2026); Liu et al. (2026c); Banerjee et al. (2026); Liu et al. (2026d); Huang and Huang (2025); Qu et al. (2026); Sadhuka et al. (2025); Zhang et al. (2026b); Weng et al. (2026); Zhu et al. (2026a); Xu et al. (2025a).

Domain-adjacent benchmark and tool instruments.

Cover biology and drug discovery (assay, image, molecular-property, and protein-benchmark checks), chemistry and materials (robotic synthesis, characterization, and property-prediction harnesses), physics and astronomy (numerical reproduction, simulator traces, observational comparison, atmospheric and astrophysical

modelling, computational fluid dynamics, and simulation-experiment instruments), medicine (conformal checks, executable pipelines, tool-service interfaces), and engineering, scientific-coding, and simulation systems (AutoML harnesses, scientific-coding benchmarks, optimization agents, simulation correctness checks, and discovery-style benchmarks) Martinek et al. (2025); Zhang et al. (2026g); Lee et al. (2025); Burgess et al. (2025); Hu et al. (2025a); Roohani et al. (2024); Liu et al. (2024); Yang et al. (2025b); Gao et al. (2025a); Flotho et al. (2025); Shi et al. (2026b); Pratihar et al. (2023); Ni et al. (2024); Li et al. (2024); Narayanan et al. (2024); Misawa et al. (2025); Rothfarb et al. (2025); Zhao et al. (2025a); Feng et al. (2026); Weidener et al. (2026b); Borrett et al. (2026); Ting et al. (2026); Zhang et al. (2026d); Islam et al. (2026); Xu and Borrett (2026); Miao et al. (2026); Mishra et al. (2025); Fox and Fox (2026); Trirat et al. (2024); Zhang et al. (2025c); Song et al. (2025); Jiang et al. (2025c); Thind et al. (2025); Kon et al. (2025); Edwards et al. (2025); Zhao et al. (2025b); Du et al. (2026a); Mohammadzadeh et al. (2025); Mudur et al. (2025); Majumder et al. (2024); Tian et al. (2024); Jing et al. (2024).

Appendix E Benchmark and Domain Evidence Notes

This appendix supports Section 4.2, Section 4.4, and Table 4. It keeps benchmark and domain details in register form keyed to the body structure.

E.1 Benchmark target notes

The benchmark notes below are organized under the four body targets, namely reconstruction, optimization, process quality, and soundness. Adjacent substrate benchmarks are included only when they explain why a target is visible or invisible to current evaluators.

Table E4: Benchmark evidence notes. The first column points back to the body target, and subclasses are supporting detail for that target.

Body target	Supporting subclass	Representative instruments	Strongest visible signal	What remains outside the signal
Reconstruction	Replication and code reconstruction	MLReplicate Gaddipati et al. (2026), PaperBench Starace et al. (2025), SciReplicate-Bench Xiang et al. (2025), PaperRepro Li et al. (2026b), CORE-Bench Siegel et al. (2024)	Executed code, reproduced results, author rubrics, or human inspection	Full claim support and interpretation
Optimization	Objective-driven search	MLE-Bench Chan et al. (2024), RE-Bench Wijk et al. (2024), MLGym Nathani et al. (2025), HeuriGym Chen et al. (2025a), KompeteAI Kulibaba et al. (2025), FML-Bench v2 Zou et al. (2026)	Competition metric, benchmark score, validation loss, or task reward	Explanation, novelty, and scope of the surrounding paper
Process quality	Research-process and literature quality	ResearchArena Zhang et al. (2026h), AutoResearchBench Xiong et al. (2026), DeepScholar-Bench Patel et al. (2025), DeepResearch Bench Du et al. (2025), EngTrace Gull et al. (2025), DR^3-Eval Xie et al. (2026a), ATLAS Liu et al. (2025a)	Review scores, retrieval metrics, trace alignment, or sandboxed report scores	Judge reliability and release authority
Soundness	Claim, citation, and integrity failures	MLR-Bench Chen et al. (2025b), SoundnessBench Ho et al. (2026), SPOT Son et al. (2025), FLAWS Xi et al. (2025), CiteTracer Li et al. (2026c), SciIntegrity-Bench Yang et al. (2026c), BadScientist Jiang et al. (2025a), DeployBench Wang et al. (2026c)	Human-labelled soundness, confirmed defects, citation checks, integrity traps, or hidden deployment checks	A default full-paper gate that blocks unsupported claims
Process quality	Tool and MCP ecosystems	MCP-Universe Luo et al. (2025c), MCPToolBench++ Fan et al. (2025), MCPAgentBench Liu et al. (2025c), MCPVerse Lei et al. (2025)	Format, static, dynamic, or real-server tool checks	Whether the final scientific conclusion follows from the tool result

Body target	Supporting subclass	Representative instruments	Strongest visible signal	What remains outside the signal
Process quality	Memory and state tracking	MEMTRACK Deshpande et al. (2025), MemoryArena He et al. (2026b), EvoMemBench Wang et al. (2026f), A-MEM Xu et al. (2025b), LEGOMem Han et al. (2025a)	Scenario success, retrieval behavior, and repeated-state diagnostics	Provenance-safe reuse of failures and counterevidence
Process quality	Human interaction and deployed behavior	KAMI Roig (2025), NORA Zhou et al. (2026a), HumanLM Wu et al. (2026c), RealUserSim Zhu et al. (2026b), ICLR review trial Thakkar et al. (2025)	Deployed conversations, expert scores, user simulation, or randomized review effects	Stable human authority over release decisions
Process quality	Literature discovery and deep research	BrowseComp Wei et al. (2025a), BrowseComp-Plus Chen et al. (2025e), DeepResearchEval Wang et al. (2026e), AstaBench Bragg et al. (2025), DeepScholar-Bench Patel et al. (2025)	Retrieval success, citation checks, and report-quality scores	New experimental or theoretical claim validity
Process quality / soundness	Rubric methodology	ResearchRubrics Sharma et al. (2025), OpenRubrics Liu et al. (2025b), CriterAlign Li et al. (2026g), RubricBench Zhang et al. (2026f)	Human-written or aligned criteria and rubric-compliance scores	Evidence sufficiency when the rubric rewards polish
Reconstruction / optimization	SWE-adjacent verifiable domains	SWE-Bench/SWE-agent Jimenez et al. (2024); Yang et al. (2024), Cybench Zhang et al. (2024), BountyBench Zhang et al. (2025a), LLM-SRBench Shojaee et al. (2025)	Tests, patches, exploits, defenses, or symbolic-regression scores	Untested behavior and scientific interpretation
Process quality	Multimodal and domain-specific tasks	MicroVQA Burgess et al. (2025), SciVisAgentBench Ai et al. (2026), MAC Jiang et al. (2025b)	Image, visualization, or multimodal scientific-question scores	Data-quality failures and claims outside the evaluated modality
Soundness / validity	Safety, security, and containment	AIxCC Zhang et al. (2026a), AgentVigil Wang et al. (2025c), safety-at-scale surveys Ma et al. (2025)	Exploit, compromise, refusal, or containment behavior	Scientific soundness of the produced research artifact
Process quality / optimization	Discovery-to-application pipelines	SciCrafter Zhou et al. (2026c), NovelSeek Zhang et al. (2025b), InternAgent-style evaluations Feng et al. (2026)	Functional checkers or end-to-end artifact delivery	Adversarial review of the story attached to the artifact
Validity threat	Contamination and disclosure	Contamination audits Musawi and Lu (2025); Bean et al. (2025), beyond-benchmark studies Srivastava et al. (2025); Abram (2026); Dussert (2026), reporting audits Zubiaga et al. (2026); Moghadasi and Ghaderi (2026)	Leakage, reporting, or benchmark-disclosure measurements	Whether a high score transfers to a new release setting

Across these notes, the same distinction recurs, where an instrument can make a failure visible without giving that failure release authority. Section 4.6 uses this distinction to define the release gap.

E.2 Domain-readout register

The domain register supports Table 4. Each row records the local readout that gives a domain verifier force and the paper-level claim type that remains outside that local readout.

Table E5: Domain-readout register. Domain rows are evidence communities for Section 4.4.

Domain	Representative evidence	Local readout	What the readout verifies	What remains outside the gate
Biology / drug discovery	Robin Ghareeb et al. (2025), Co-Scientist Gottweis et al. (2025), AutoScientists Gao et al. (2026), Agentomics-ML Martinek et al. (2025), DrugSAGE Zhang et al. (2026g)	Assay, RNA-seq, protein or molecular-property benchmark	Candidate support, classification or property performance, selected biological signal	Mechanism, clinical relevance, and wet-lab generalization
Chemistry / materials	A-Lab GPSS Fei et al. (2026), MatBench Dunn et al. (2020), ElementsClaw Li et al. (2026d), DiscoveryBench Majumder et al. (2024)	Robotic synthesis, characterization, curated property score, data-grounded discovery task	Sample, composition, property, or benchmark answer	Mechanism, synthesizability outside the campaign, and paper narrative
Physics / astronomy	PhysMaster Miao et al. (2025b), grounded computational physics Huang (2026), Kosmos Mitchener et al. (2025b), TianJi Zhang et al. (2026d), GWAgent Islam et al. (2026)	Recomputed quantitative claims, simulator traces, observational comparison, code diagnostics	Numeric agreement, simulation output, or statement-level support	Model adequacy, missing failure cases, and interpretation
Medicine / clinical research	Medical AI Scientist Wu et al. (2026b), NORA Zhou et al. (2026a), iatroX Tytler (2025), ConfAgents Zhao et al. (2025a), ClawdLab Weidener et al. (2026b)	Executed medical-AI tasks, expert checkpoints, field-use feedback, conformal diagnosis	Workflow output, case-level score, or deployment feedback	Patient outcomes, safety under clinical action, and human accountability
Engineering / computer science	AlphaEvolve Novikov et al. (2025), CORAL Qu et al. (2026), R&D-Agent Yang et al. (2025a), SWE-Bench Jimenez et al. (2024), BountyBench Zhang et al. (2025a), DeployBench Wang et al. (2026c)	Tests, hidden graders, competition metrics, patches, deployment oracles	Typed artifact behavior under a scorer	Causal explanation, scope, scientific importance, and paper-level claims
Quantum	AI-Mandel Arlt et al. (2025), QAgent Fu et al. (2025), Qumus Shi et al. (2026a)	Circuit execution, Qiskit-style tests, photonic tools, device fabrication	Circuit, simulator, or device-level behavior	Novelty, interpretation, and prose around the circuit or device

The body conclusion follows from the table rather than from any one domain, since local ground truth can be strong while still failing to adjudicate every central claim in a generated paper.

Appendix F Gap-map Coding Notes

This appendix supports Table 6. The heatmap summarizes release-relevant functions discussed in Section 4, while system-axis labels remain in Table 2.

Cell codes.

S means that at least one cited public artifact gives strong evidence for the function at the row's stated scope. **P** means that the evidence is partial, local, or lacks release authority. **0** means that the function is absent from the reported public design or not measured in a way that supports the function. A 0 does not mean the artifact could never be adapted for that function.

Function	Strong evidence (S)	Partial evidence (P)	Absent evidence (0)
Soundness	Claims are checked against declared evidence with reported audit results.	Selected numbers, citations, or local claims are checked.	Outputs are scored or reviewed without evidence-support testing.
Hard verification	A non-preference gate can reject a typed object.	A gate checks a narrow object or is advisory.	Judgment is non-gating review or self-critique only.
Adversarial review	An independent actor or tool searches for counterevidence and can affect the outcome.	Review is separated by role, model, or trace access but lacks full authority.	The generator reviews its own output or receives style feedback only.
Usage-grounded evaluation	Evidence comes from deployed or realistic use with users, tools, traces, or downstream artifacts.	The study approximates use through scenarios, simulations, or small expert cases.	Only static benchmark or paper-review scores are reported.
Loop architecture	Failed evidence, objections, or hidden checks change later actions or release decisions.	Feedback changes retries or local search but not the full claim lifecycle.	The loop optimizes a score or draft without contradiction state.
Human authority	Human responsibility, evidence access, and veto points are explicit.	Humans steer, score, or approve selected steps.	Human involvement is absent, ceremonial, or unspecified.

Row scope.

Rows in Table 6 are representative evidence clusters, not exhaustive literature classes. For benchmark rows, S means the benchmark strongly measures the function, and does not mean the benchmark supplies a production mechanism. For system rows, S requires a reported mechanism at the relevant scope, not merely a plausible adaptation.

References

Abram M (2026) Toward evaluation frameworks for multi-agent scientific ai systems. URL https://arxiv.org/abs/2603.26718, arXiv:2603.26718

Abramovich T, Udeshi M, Shao M, et al (2024) Enigma: Interactive tools substantially assist lm agents in finding security vulnerabilities. URL https://arxiv.org/abs/2409.16165, arXiv:2409.16165

Agarwal A (2026) Refute-or-promote: An adversarial stage-gated multi-agent review methodology for high-precision llm-assisted defect discovery. URL https://arxiv.org/abs/2604.19049, arXiv:2604.19049

Agarwal S, Sahu G, Puri A, et al (2024) Litllms, llms for literature review: Are we there yet? URL https://arxiv.org/abs/2412.15249, arXiv:2412.15249

Ai K, Miao H, Tang K, et al (2026) Scivisagentbench: A benchmark for evaluating scientific data analysis and visualization agents. URL https://arxiv.org/abs/2603.29139, arXiv:2603.29139

Alzahrani AH (2026) Persistent ai agents in academic research: A single-investigator implementation case study. URL https://arxiv.org/abs/2605.26870, arXiv:2605.26870

Ambati M (2025) Proofnet++: A neuro-symbolic system for formal proof verification with self-correction. URL https://arxiv.org/abs/2505.24230, arXiv:2505.24230

Anurin A, Ng J, Schaffer K, et al (2024) Catastrophic cyber capabilities benchmark (3cb): Robustly evaluating llm agent cyber offense capabilities. URL https://arxiv.org/abs/2410.09114, arXiv:2410.09114

Arlt S, Gu X, Krenn M (2025) Towards autonomous quantum physics research using llm agents with access to intelligent tools. URL https://arxiv.org/abs/2511.11752, arXiv:2511.11752

Asai A, He J, Shao R, et al (2024) Openscholar: Synthesizing scientific literature with retrieval-augmented lms. URL https://arxiv.org/abs/2411.14199, arXiv:2411.14199

Baek J, Jauhar S, Cucerzan S, et al (2024) Researchagent: Iterative research idea generation over scientific literature with large language models. URL https://arxiv.org/abs/2404.07738, arXiv:2404.07738

Banerjee D, Xu C, Ie E, et al (2026) Severa: Verified synthesis of self-evolving agents. URL https://arxiv.org/abs/2603.25111, arXiv:2603.25111

Bean AM, Kearns R, Romanou A, et al (2025) Measuring what matters: Construct validity in large language model benchmarks. URL https://arxiv.org/abs/2511.04703, arXiv:2511.04703

Beel J, Kan MY, Baumgart M (2025) Evaluating sakana's ai scientist: Bold claims, mixed results, and a promising future? URL https://arxiv.org/abs/2502.14297, arXiv:2502.14297

Bengio Y, Clare S, Prunkl C, et al (2026) International ai safety report 2026. URL https://arxiv.org/abs/2602.21012, arXiv:2602.21012

Borrett T, Xu L, Nilipour A, et al (2026) Competing with ai scientists: Agent-driven approach to astrophysics research. URL https://arxiv.org/abs/2604.09621, arXiv:2604.09621

Bragg J, D'Arcy M, Balepur N, et al (2025) Astabench: Rigorous benchmarking of ai agents with a scientific research suite. URL https://arxiv.org/abs/2510.21652, arXiv:2510.21652

Burgess J, Nirschl JJ, Bravo-Sanchez L, et al (2025) Microvqa: A multimodal reasoning benchmark for microscopy-based scientific research. URL https://arxiv.org/abs/2503.13399, arXiv:2503.13399

Cai Z, Behl H (2026) Ai scientist via synthetic task scaling. URL https://arxiv.org/abs/2603.17216, arXiv:2603.17216

Cao H, Liu S, Zhang F, et al (2026) Mozi: Governed autonomy for drug discovery llm agents. URL https://arxiv.org/abs/2603.03655, arXiv:2603.03655

Chan JS, Chowdhury N, Jaffe O, et al (2024) Mle-bench: Evaluating machine learning agents on machine learning engineering. URL https://arxiv.org/abs/2410.07095, arXiv:2410.07095

Chang S, Anderson A, Hofman JM (2025) Chatbench: From static benchmarks to human-ai evaluation. URL https://arxiv.org/abs/2504.07114, arXiv:2504.07114

Chen H, Wang Y, Cai Y, et al (2025a) Heurigym: An agentic benchmark for llm-crafted heuristics in combinatorial optimization. URL https://arxiv.org/abs/2506.07972, arXiv:2506.07972

Chen H, Xiong M, Lu Y, et al (2025b) Mlr-bench: Evaluating ai agents on open-ended machine learning research. URL https://arxiv.org/abs/2505.19955, arXiv:2505.19955

Chen R (2025) Evidence-bound autonomous research (evibound): A governance framework for eliminating false claims. URL https://arxiv.org/abs/2511.05524, arXiv:2511.05524

Chen T, Lin B, Anumasa S, et al (2025c) Auto-discovery-bench: Diagnosing structured state tracking in oracle-guided discovery. URL https://arxiv.org/abs/2502.15224, arXiv:2502.15224

Chen Y, Lv T, Zhang S, et al (2025d) Paper2web: Let's make your paper alive! URL https://arxiv.org/abs/2510.15842, arXiv:2510.15842

Chen Z, Ma X, Zhuang S, et al (2025e) Browsecomp-plus: A more fair and transparent evaluation benchmark of deep-research agent. URL https://arxiv.org/abs/2508.06600, arXiv:2508.06600

Cheng J, Kumar A, Lal R, et al (2025) Webatlas: An llm agent with experience-driven memory and action simulation. URL https://arxiv.org/abs/2510.22732, arXiv:2510.22732

Chervonyi Y, Trinh TH, Olšák M, et al (2025) Gold-medalist performance in solving olympiad geometry with alphageometry2. URL https://arxiv.org/abs/2502.03544, arXiv:2502.03544

Chhikara P, Khant D, Aryan S, et al (2025) Mem0: Building production-ready ai agents with scalable long-term memory. URL https://arxiv.org/abs/2504.19413, arXiv:2504.19413

Deng X, Da J, Pan E, et al (2025) Swe-bench pro: Can ai agents solve long-horizon software engineering tasks? URL https://arxiv.org/abs/2509.16941, arXiv:2509.16941

Deshpande D, Gangal V, Mehta H, et al (2025) Memtrack: Evaluating long-term memory and state tracking in multi-platform dynamic agent environments. URL https://arxiv.org/abs/2510.01353, arXiv:2510.01353

Dong Z, Du S, Lu Z, et al (2025) Cfd-copilot: leveraging domain-adapted large language model and model context protocol to enhance simulation automation. URL https://arxiv.org/abs/2512.07917, arXiv:2512.07917

Dou Y, Galley M, Peng B, et al (2025) Simulatorarena: Are user simulators reliable proxies for multi-turn evaluation of ai assistants? URL https://arxiv.org/abs/2510.05444, arXiv:2510.05444

Du J, Sun Y, Yang H (2026a) AutoNumerics: An autonomous, pde-agnostic multi-agent pipeline for scientific computing. URL https://arxiv.org/abs/2602.17607, arXiv:2602.17607

Du M, Xu B, Zhu C, et al (2025) Deepresearch bench: A comprehensive benchmark for deep research agents. URL https://arxiv.org/abs/2506.11763, arXiv:2506.11763

Du Y, Yang X, Zhou Z, et al (2026b) Datamaster: Data-centric autonomous ai research. URL https://arxiv.org/abs/2605.10906, arXiv:2605.10906

Dunn A, Wang Q, Ganose A, et al (2020) Benchmarking materials property prediction methods: The matbench test set and automatminer reference algorithm. URL https://arxiv.org/abs/2005.00707, arXiv:2005.00707

Dussert J (2026) Who judges the judges? governance from metrics: a runtime framework for continuous llm compliance monitoring. URL https://arxiv.org/abs/2605.24737, arXiv:2605.24737

Edwards N, Lee Y, Mao Y, et al (2025) Rexbench: Can coding agents autonomously implement ai research extensions? URL https://arxiv.org/abs/2506.22598, arXiv:2506.22598

Egg A, Goyanes MI, Kingma F, et al (2025) Dabstep: Data agent benchmark for multi-step reasoning. URL https://arxiv.org/abs/2506.23719, arXiv:2506.23719

Eriksson M, Purificato E, Noroozian A, et al (2025) Can we trust ai benchmarks? an interdisciplinary review of current issues in ai evaluation. URL https://arxiv.org/abs/2502.06559, arXiv:2502.06559

Fan SJ, Ding X, Zhang L, et al (2025) Mcptoolbench++: A large scale ai agent model context protocol mcp tool use benchmark. URL https://arxiv.org/abs/2508.07575, arXiv:2508.07575

Fang J, Peng Y, Zhang X, et al (2025) A comprehensive survey of self-evolving ai agents. arXiv:2508.07407

Fang S, Ding W, Cao Z, et al (2026) Aegis: From clues to verdicts – graph-guided deep vulnerability reasoning via dialectics and meta-auditing. URL https://arxiv.org/abs/2603.20637, arXiv:2603.20637

Fei Y, Rendy B, Yang X, et al (2026) Agentic llm reasoning in a self-driving laboratory for air-sensitive lithium halide spinel conductors. URL https://arxiv.org/abs/2604.11957, arXiv:2604.11957

Feng J, Huang S, Qu X, et al (2025) Retool: Reinforcement learning for strategic tool use in llms. URL https://arxiv.org/abs/2504.11536, arXiv:2504.11536

Feng S, Ma R, Yan X, et al (2026) Internagent-1.5: A unified agentic framework for long-horizon autonomous scientific discovery. URL https://arxiv.org/abs/2602.08990, arXiv:2602.08990

Flotho M, Diks IF, Flotho P, et al (2025) Mcpmed: A call for mcp-enabled bioinformatics web services for llm-driven discovery. URL https://arxiv.org/abs/2507.08055, arXiv:2507.08055

Fox J, Fox GC (2026) Experiments in agentic ai for science. URL https://arxiv.org/abs/2605.26305, arXiv:2605.26305

Fu Z, Chen F, Jiang L (2025) Qagent: An llm-based multi-agent system for autonomous openqasm programming. URL https://arxiv.org/abs/2508.20134, arXiv:2508.20134

Gaddipati SK, Muhammed D, Keya F, et al (2026) Mlreplicate: Benchmarking autonomous research systems for machine learning reproducibility. URL https://arxiv.org/abs/2605.16616, arXiv:2605.16616

Gao S, Zhu R, Sui P, et al (2025a) Democratizing ai scientists using tooluniverse. URL https://arxiv.org/abs/2509.23426, arXiv:2509.23426

Gao S, Fang A, Zitnik M (2026) Autoscientists: Self-organizing agent teams for long-running scientific experimentation. URL https://arxiv.org/abs/2605.28655, arXiv:2605.28655

Gao X, Xie S, Zhai J, et al (2025b) Mcp-radar: A multi-dimensional benchmark for evaluating tool use capabilities in large language models. URL https://arxiv.org/abs/2505.16700, arXiv:2505.16700

Geng X, Xia P, Zhang Z, et al (2025) Webwatcher: Breaking new frontier of vision-language deep research agent. URL https://arxiv.org/abs/2508.05748, arXiv:2508.05748

Ghareeb AE, Chang B, Mitchener L, et al (2025) Robin: A multi-agent system for automating scientific discovery. URL https://arxiv.org/abs/2505.13400, arXiv:2505.13400

Goel S, Hazra R, Jayalath D, et al (2025) Training ai co-scientists using rubric rewards. URL https://arxiv.org/abs/2512.23707, arXiv:2512.23707

Google DeepMind (2024) Ai achieves silver-medal standard solving international mathematical olympiad problems. URL https://deepmind.google/blog/ai-solves-imo-problems-at-silver-medal-level/, accessed 2026-06-09

Gottweis J, Weng WH, Daryin A, et al (2025) Towards an ai co-scientist. URL https://arxiv.org/abs/2502.18864, arXiv:2502.18864

Goyal P, Parmar M, Song Y, et al (2026) Scholarpeer: A context-aware multi-agent framework for automated peer review. URL https://arxiv.org/abs/2601.22638, arXiv:2601.22638

Gu J, Jiang X, Shi Z, et al (2024) A survey on llm-as-a-judge. URL https://arxiv.org/abs/2411.15594, arXiv:2411.15594

Gu Y, Yang J, Luo J, et al (2026) Scicustom: A framework for custom evaluation of scientific capabilities in large language models. URL https://arxiv.org/abs/2605.19357, arXiv:2605.19357

Gull A, Safder MU, Elbadry R, et al (2025) Engtrace: A symbolic benchmark for verifiable process supervision of engineering reasoning. URL https://arxiv.org/abs/2511.01650, arXiv:2511.01650

Gunjal A, Wang A, Lau E, et al (2025) Rubrics as rewards: Reinforcement learning beyond verifiable domains. URL https://arxiv.org/abs/2507.17746, arXiv:2507.17746

Guo Y, Liu P, Ma W, et al (2025) Mcpxkit: the unified toolkit for analyzing model context protocol security. URL https://arxiv.org/abs/2508.12538, arXiv:2508.12538

Gupta T, Pruthi D (2025) All that glitters is not novel: Plagiarism in ai generated research. URL https://arxiv.org/abs/2502.16487, arXiv:2502.16487

Haddad D, Bareket D, Chang J, et al (2026) Understanding usage and engagement in ai-powered scientific research tools: The asta interaction dataset. URL https://arxiv.org/abs/2602.23335, arXiv:2602.23335

Hambardzumyan K, Baldwin N, Toledo E, et al (2026) Aira_2: Overcoming bottlenecks in ai research agents. URL https://arxiv.org/abs/2603.26499, arXiv:2603.26499

Han D, Couturier C, Diaz DM, et al (2025a) Legomem: Modular procedural memory for multi-agent llm systems for workflow automation. URL https://arxiv.org/abs/2510.04851, arXiv:2510.04851

Han J, Kim H, Lee C, et al (2025b) Deer: A benchmark for evaluating deep research agents on expert report generation. URL https://arxiv.org/abs/2512.17776, arXiv:2512.17776

Handoko A, Made RI (2025) Artificial intelligence and generative models for materials discovery – a review. URL https://arxiv.org/abs/2508.03278, arXiv:2508.03278

He P, Dai Z, He B, et al (2025) Traject-bench:a trajectory-aware benchmark for evaluating agentic tool use. URL https://arxiv.org/abs/2510.04550, arXiv:2510.04550

He S, Chen Z, Zhao Y, et al (2026a) From language to action: Enhancing llm task efficiency with task-aware mcp server recommendation. URL https://arxiv.org/abs/2604.17234, arXiv:2604.17234

He Z, Wang Y, Zhi C, et al (2026b) Memoryarena: Benchmarking agent memory in interdependent multi-session agentic tasks. URL https://arxiv.org/abs/2602.16313,

arXiv:2602.16313

Hirota W, Taniguchi T, Ohkuma T, et al (2026) Aggregate vs. personalized judges in business idea evaluation: Evidence from expert disagreement. URL https://arxiv.org/abs/2604.22517, arXiv:2604.22517

Ho ST, Liu M, Nghiem H, et al (2026) Soundnessbench: Can your ai scientist really tell good research ideas from bad ones? URL https://arxiv.org/abs/2605.30329, arXiv:2605.30329

Hong H, Li Y, Chen J, et al (2026) Hiras: A hierarchical multi-agent framework for paper-to-code generation and execution. URL https://arxiv.org/abs/2604.17745, arXiv:2604.17745

Hu B, Park JH, Chen H, et al (2025a) Domain knowledge infused conditional generative models for accelerating drug discovery. URL https://arxiv.org/abs/2510.09837, arXiv:2510.09837

Hu C, Zhang L, Lim Y, et al (2025b) Repro-bench: Can agentic ai systems assess the reproducibility of social science research? URL https://arxiv.org/abs/2507.18901, arXiv:2507.18901

Hu M, Ma C, Li W, et al (2025c) A survey of scientific large language models: From data foundations to agent frontiers. URL https://arxiv.org/abs/2508.21148, arXiv:2508.21148

Hu S, Lu C, Clune J (2024) Automated design of agentic systems. URL https://arxiv.org/abs/2408.08435, arXiv:2408.08435

Hua T, Hua H, Xiang V, et al (2025) Researchcodebench: Benchmarking llms on implementing novel machine learning research code. URL https://arxiv.org/abs/2506.02314, arXiv:2506.02314

Huang H (2026) Towards grounded autonomous research: an end-to-end llm mini research loop on published computational physics. URL https://arxiv.org/abs/2604.12198, arXiv:2604.12198

Huang J, Cucerzan S, Jauhar SK, et al (2025a) Idea2plan: Exploring ai-powered research planning. URL https://arxiv.org/abs/2510.24891, arXiv:2510.24891

Huang K, Huang J (2025) Audited skill-graph self-improvement for agentic llms via verifiable rewards, experience synthesis, and continual memory. URL https://arxiv.org/abs/2512.23760, arXiv:2512.23760

Huang P, Zhong Z, Wan Z, et al (2026) Mmdeepresearch-bench: A benchmark for multimodal deep research agents. URL https://arxiv.org/abs/2601.12346, arXiv:2601.12346

Huang Y, Chen Y, Zhang H, et al (2025b) Deep research agents: A systematic examination and roadmap. URL https://arxiv.org/abs/2506.18096, arXiv:2506.18096

Hwang JD, Kishore V, Singh A, et al (2026) Deep research, shallow evaluation: A case study in meta-evaluation for long-form qa benchmarks. URL https://arxiv.org/abs/2603.06942, arXiv:2603.06942

Islam T, Wadekar D, Venumadhav T, et al (2026) Discovery of interpretable surrogates via agentic ai: Application to gravitational waves. URL https://arxiv.org/abs/2605.11280, arXiv:2605.11280

Jiang B (2026) Hindsight: Evaluating llm-generated research ideas via future impact. URL https://arxiv.org/abs/2603.15164, arXiv:2603.15164

Jiang F, Feng Y, Li Y, et al (2025a) Badscientist: Can a research agent write convincing but unsound papers that fool llm reviewers? URL https://arxiv.org/abs/2510.18003, arXiv:2510.18003

Jiang M, Gao J, Zhan J, et al (2025b) Mac: A live benchmark for multimodal large language models in scientific understanding. URL https://arxiv.org/abs/2508.15802, arXiv:2508.15802

Jiang Z, Schmidt D, Srikanth D, et al (2025c) Aide: Ai-driven exploration in the space of code. URL https://arxiv.org/abs/2502.13138, arXiv:2502.13138

Jimenez CE, Yang J, Wettig A, et al (2024) Swe-bench: Can language models resolve real-world github issues? URL https://arxiv.org/abs/2310.06770, arXiv:2310.06770

Jing L, Huang Z, Wang X, et al (2024) Dsbench: How far are data science agents from becoming data science experts? URL https://arxiv.org/abs/2409.07703, arXiv:2409.07703

Kargupta P, Agarwal I, August T, et al (2025) Tree-of-debate: Multi-persona debate trees elicit critical thinking for scientific comparative analysis. URL https://arxiv.org/abs/2502.14767, arXiv:2502.14767

Ke Y, George K, Pandya K, et al (2025) Biodisco: Multi-agent hypothesis generation with dual-mode evidence, iterative feedback and temporal evaluation. URL https://arxiv.org/abs/2508.01285, arXiv:2508.01285

Keuper J (2025) Prompt injection attacks on llm generated reviews of scientific publications. URL https://arxiv.org/abs/2509.10248, arXiv:2509.10248

Kim G, Wilf A, Morency LP, et al (2025) From reproduction to replication: Evaluating research agents with progressive code masking. URL https://arxiv.org/abs/2506.19724, arXiv:2506.19724

Kon PTJ, Liu J, Zhu X, et al (2025) Exp-bench: Can ai conduct ai research experiments? URL https://arxiv.org/abs/2505.24785, arXiv:2505.24785

Kulibaba S, Dzhalilov A, Pakhomov R, et al (2025) Kompeteai: Accelerated autonomous multi-agent system for end-to-end pipeline generation for machine learning problems. URL https://arxiv.org/abs/2508.10177, arXiv:2508.10177

Lavi G (2026) Right-to-act: A pre-execution non-compensatory decision protocol for ai systems. URL https://arxiv.org/abs/2604.24153, arXiv:2604.24153

Lee N, Brouwer ED, Hajiramezanali E, et al (2025) Rag-enhanced collaborative llm agents for drug discovery. URL https://arxiv.org/abs/2502.17506, arXiv:2502.17506

Lei F, Yang Y, Sun W, et al (2025) Mcpverse: An expansive, real-world benchmark for agentic tool use. URL https://arxiv.org/abs/2508.16260, arXiv:2508.16260

Li H, Shi Y, Lin S, et al (2025a) Swe-debate: Competitive multi-agent debate for software issue resolution. URL https://arxiv.org/abs/2507.23348, arXiv:2507.23348

Li L, Li D, Chen C, et al (2026a) Llm-as-a-judge in healthcare: A scoping analysis of applications, methods, and human alignment. URL https://arxiv.org/abs/2605.25273, arXiv:2605.25273

Li L, Wang R, Song H, et al (2026b) What papers don't tell you: Recovering tacit knowledge for automated paper reproduction. URL https://arxiv.org/abs/2603.01801, arXiv:2603.01801

Li M, Lin Z, Ma S (2026c) Source or it didn't happen: A multi-agent framework for citation hallucination detection. URL https://arxiv.org/abs/2605.08583, arXiv:2605.08583

Li M, Rong Y, Li S, et al (2026d) Agentic fusion of large atomic and language models to accelerate superconductor discovery. URL https://arxiv.org/abs/2604.23758, arXiv:2604.23758

Li N (2026a) The ideation bottleneck: Decomposing the quality gap between ai-generated and human economics research. URL https://arxiv.org/abs/2604.03338, arXiv:2604.03338

Li R, Jing L, Han C, et al (2024) Ldc: Learning to generate research idea with dynamic control. URL https://arxiv.org/abs/2412.14626, arXiv:2412.14626

Li R, Gu JC, Kung PN, et al (2025b) Llm-reval: Can we trust llm reviewers yet? URL https://arxiv.org/abs/2510.12367, arXiv:2510.12367

Li R, Tu J, Su Y, et al (2025c) Toolrm: Towards agentic tool-use reward modeling. URL https://arxiv.org/abs/2510.26167, arXiv:2510.26167

Li R, Du M, Xu B, et al (2026e) Deepresearch bench ii: Diagnosing deep research agents via rubrics from expert report. URL https://arxiv.org/abs/2601.08536, arXiv:2601.08536

Li X (2026b) Auto researching, not hyperparameter tuning: Convergence analysis of 10,000 experiments. URL https://arxiv.org/abs/2603.15916, arXiv:2603.15916

Li Y, Shao C, Liu X, et al (2026f) Autosota: An end-to-end automated research system for state-of-the-art ai model discovery. URL https://arxiv.org/abs/2604.05550, arXiv:2604.05550

Li Z, Zhang H, Han S, et al (2025d) In-the-flow agentic system optimization for effective planning and tool use. URL https://arxiv.org/abs/2510.05592, arXiv:2510.05592

Li Z, Cvejić A, Chen Z, et al (2026g) Criteralign: Criterion-centric rationale alignment for code preference judging. URL https://arxiv.org/abs/2605.19665, arXiv:2605.19665

Li Z, Jiang D, Ma X, et al (2026h) Openresearcher: A fully open pipeline for long-horizon deep research trajectory synthesis. URL https://arxiv.org/abs/2603.20278, arXiv:2603.20278

Liang Y, Chen X, Ge Y, et al (2026) Unitoolcall: Unifying tool-use representation, data, and evaluation for llm agents. URL https://arxiv.org/abs/2604.11557, arXiv:2604.11557

Lin H, Tan X, Qin Y, et al (2025) Cuarewardbench: A benchmark for evaluating reward models on computer-using agent. URL https://arxiv.org/abs/2510.18596, arXiv:2510.18596

Liu H, Liu J, Liu S, et al (2025a) Atlas: A high-difficulty, multidisciplinary benchmark for frontier scientific reasoning. URL https://arxiv.org/abs/2511.14366, arXiv:2511.14366

Liu H, Choi Y, Gautam S, et al (2026a) Who owns creativity and who does the work? trade-offs in llm-supported research ideation. URL https://arxiv.org/abs/2601.12152, arXiv:2601.12152

Liu J, Pei J, Huang J, et al (2026b) The last human-written paper: Agent-native research artifacts. URL https://arxiv.org/abs/2604.24658, arXiv:2604.24658

Liu J, Qiu S, Li M, et al (2026c) Autoresearchclaw: Self-reinforcing autonomous research with human-ai collaboration. URL https://arxiv.org/abs/2605.20025, arXiv:2605.20025

Liu S, Lu Y, Chen S, et al (2024) Drugagent: Automating ai-aided drug discovery programming through llm multi-agent collaboration. URL https://arxiv.org/abs/2411.15692, arXiv:2411.15692

Liu T, Xu R, Yu TCB, et al (2025b) Openrubrics: Towards scalable synthetic rubric generation for reward modeling and llm alignment. URL https://arxiv.org/abs/2510.07743, arXiv:2510.07743

Liu W, Liu Z, Dai E, et al (2025c) Mcpagentbench: A real-world task benchmark for evaluating llm agent mcp tool use. URL https://arxiv.org/abs/2512.24565, arXiv:2512.24565

Liu X, Kodakandla D, Srivastva K, et al (2026d) Veritrans: Fine-tuned llm-assisted nl-to-pl translation via a deterministic neuro-symbolic pipeline. URL https://arxiv.org/abs/2604.10341, arXiv:2604.10341

Liu X, Zhang TJ, Schölkopf B, et al (2026e) Stargazer: A scalable model-fitting benchmark environment for ai agents under astrophysical constraints. URL https://arxiv.org/abs/2604.15664, arXiv:2604.15664

Liu Y, Yang Z, Xie T, et al (2025d) Researchbench: Benchmarking llms in scientific discovery via inspiration-based task decomposition. URL https://arxiv.org/abs/2503.21248, arXiv:2503.21248

Loosmore S (2026) Leverage laws: A per-task framework for human-agent collaboration. URL https://arxiv.org/abs/2604.25040, arXiv:2604.25040

Lu C, Lu C, Lange R, et al (2024) The ai scientist: Towards fully automated open-ended scientific discovery. URL https://arxiv.org/abs/2408.06292, arXiv:2408.06292

Lumer E, Nizar F, Gulati A, et al (2025) Tool-to-agent retrieval: Bridging tools and agents for scalable llm multi-agent systems. URL https://arxiv.org/abs/2511.01854, arXiv:2511.01854

Luo Y, Yu Z, Wang X, et al (2025a) What makes ai research replicable? executable knowledge graphs as scientific knowledge representations. URL https://arxiv.org/abs/2510.17795, arXiv:2510.17795

Luo Z, Kasirzadeh A, Shah NB (2025b) The more you automate, the less you see: Hidden pitfalls of ai scientist systems. URL https://arxiv.org/abs/2509.08713, arXiv:2509.08713

Luo Z, Shen Z, Yang W, et al (2025c) Mcp-universe: Benchmarking large language models with real-world model context protocol servers. URL https://arxiv.org/abs/2508.14704, arXiv:2508.14704

Lyu Y, Zhang X, Yi X, et al (2026) Evoscientist: Towards multi-agent evolving ai scientists for end-to-end scientific discovery. URL https://arxiv.org/abs/2603.08127, arXiv:2603.08127

Ma X, Gao Y, Wang Y, et al (2025) Safety at scale: A comprehensive survey of large model and agent safety. URL https://arxiv.org/abs/2502.05206, arXiv:2502.05206

Mahmoud A, Rezaei M, Wang Z, et al (2026) Reward hacking in rubric-based reinforcement learning. URL https://arxiv.org/abs/2605.12474, arXiv:2605.12474

Majumder BP, Surana H, Agarwal D, et al (2024) Discoverybench: Towards data-driven discovery with large language models. URL https://arxiv.org/abs/2407.01725, arXiv:2407.01725

Malka J, Zacchiroli S, Zimmermann T (2026) Docker does not guarantee reproducibility. URL https://arxiv.org/abs/2601.12811, arXiv:2601.12811

Martinek V, Gariboldi A, Tzimotoudis D, et al (2025) Agentomics-ml: Autonomous machine learning experimentation agent for genomic and transcriptomic data. URL https://arxiv.org/abs/2506.05542, arXiv:2506.05542

Maruyama N, Yoshida T, Sato H, et al (2025) A concurrent modular agent: Framework for autonomous llm agents. URL https://arxiv.org/abs/2508.19042, arXiv:2508.19042

Meng R, Mishra BD, Chen J, et al (2026) Scientistone: Towards human-level autonomous research via chain-of-evidence. URL https://arxiv.org/abs/2605.26340, arXiv:2605.26340

Miao H, Li Z, Ai K, et al (2026) Toward ai vis co-scientists: A general and end-to-end agent harness for solving complex data visualization tasks. URL https://arxiv.org/abs/2605.21825, arXiv:2605.21825

Miao J, Davis JR, Zhang Y, et al (2025a) Paper2agent: Reimagining research papers as interactive and reliable ai agents. URL https://arxiv.org/abs/2509.06917, arXiv:2509.06917

Miao T, Dai J, Liu J, et al (2025b) Physmaster: Building an autonomous ai physicist for theoretical and computational physics research. URL https://arxiv.org/abs/2512.19799, arXiv:2512.19799

Misawa T, Koizumi A, Tamura R, et al (2025) Exploring utilization of generative ai for research and education in data-driven materials science. URL https://arxiv.org/abs/2504.08817, arXiv:2504.08817

Mishra S, Sahu G, Pedersoli M, et al (2025) Ainstein: Can llms solve research problems from parametric memory alone? URL https://arxiv.org/abs/2510.05432, arXiv:2510.05432

Mitchell RJ (2026) When the agent is the adversary: Architectural requirements for agentic ai containment after the april 2026 frontier model escape. URL https://arxiv.org/abs/2604.23425, arXiv:2604.23425

Mitchener L, Laurent JM, Andonian A, et al (2025a) Bixbench: a comprehensive benchmark for llm-based agents in computational biology. URL https://arxiv.org/abs/2503.00096, arXiv:2503.00096

Mitchener L, Yiu A, Chang B, et al (2025b) Kosmos: An ai scientist for autonomous discovery. URL https://arxiv.org/abs/2511.02824, arXiv:2511.02824

Miyai A, Toyooka M, Zhao Z, et al (2026) Paper reconstruction evaluation: Evaluating presentation and hallucination in ai-written papers. URL https://arxiv.org/abs/2604.01128, arXiv:2604.01128

Moghadasi M, Ghaderi F (2026) What twelve llm agent benchmark papers disclose about themselves: A pilot audit and an open scoring schema. URL https://arxiv.org/abs/2605.21404, arXiv:2605.21404

Mohammadzadeh S, Hamdi E, Shor J, et al (2025) Fem-bench: A structured scientific reasoning benchmark for evaluating code-generating llms. URL https://arxiv.org/abs/2512.20732, arXiv:2512.20732

Mozannar H, Bansal G, Tan C, et al (2025) Magentic-ui: Towards human-in-the-loop agentic systems. URL https://arxiv.org/abs/2507.22358, arXiv:2507.22358

Mudur N, Cui H, Venugopalan S, et al (2025) Feabench: Evaluating language models on multiphysics reasoning ability. URL https://arxiv.org/abs/2504.06260, arXiv:2504.06260

Musawi R, Lu S (2025) Towards contamination resistant benchmarks. URL https://arxiv.org/abs/2505.08389, arXiv:2505.08389

Narayanan S, Braza JD, Griffiths RR, et al (2024) Aviary: training language agents on challenging scientific tasks. URL https://arxiv.org/abs/2412.21154, arXiv:2412.21154

Nathani D, Madaan L, Roberts N, et al (2025) Mlgym: A new framework and benchmark for advancing ai research agents. URL https://arxiv.org/abs/2502.14499, arXiv:2502.14499

Ni Z, Li Y, Hu K, et al (2024) Matpilot: an llm-enabled ai materials scientist under the framework of human-machine collaboration. URL https://arxiv.org/abs/2411.08063, arXiv:2411.08063

Novikov A, Vũ N, Eisenberger M, et al (2025) Alphaevolve: A coding agent for scientific and algorithmic discovery. URL https://arxiv.org/abs/2506.13131, arXiv:2506.13131

Pan H, Chard R, Mello R, et al (2025) Experiences with model context protocol servers for science and high performance computing. URL https://arxiv.org/abs/2508.18489, arXiv:2508.18489

Pan T, Ouyang J, Cheng M, et al (2026) Paperscout: An autonomous agent for academic paper search with process-aware sequence-level policy optimization. URL https://arxiv.org/abs/2601.10029, arXiv:2601.10029

Panek E, Roman A, Shukla G, et al (2026) Aster – agentic science toolkit for exoplanet research. URL https://arxiv.org/abs/2603.26953, arXiv:2603.26953

Panfilov A, Romov P, Shilov I, et al (2026) Claudini: Autoresearch discovers state-of-the-art adversarial attack algorithms for llms. URL https://arxiv.org/abs/2603.24511, arXiv:2603.24511

Panigrahi SS, Videnović J, Brbić M (2026) Heurekabench: A benchmarking framework for ai co-scientist. URL https://arxiv.org/abs/2601.01678, arXiv:2601.01678

Patel L, Arabzadeh N, Gupta H, et al (2025) Deepscholar-bench: A live benchmark and automated evaluation for generative research synthesis. URL https://arxiv.org/abs/2508.20033, arXiv:2508.20033

Pratihar SR, Pai D, Nag MVA (2023) Aims-erea - a framework for ai-accelerated innovation of materials for sustainability - for environmental remediation and energy applications. URL https://arxiv.org/abs/2311.11060, arXiv:2311.11060

Qi J, Li M, Liu J, et al (2026) Towards trustworthy agentic ai: a comprehensive survey of safety, robustness, privacy, and system security. URL https://arxiv.org/abs/2605.23989, arXiv:2605.23989

Qian C, Acikgoz EC, He Q, et al (2025) Toolrl: Reward is all tool learning needs. URL https://arxiv.org/abs/2504.13958, arXiv:2504.13958

Qian H, Liu Z (2025) Metaagent: Toward self-evolving agent via tool meta-learning. URL https://arxiv.org/abs/2508.00271, arXiv:2508.00271

Qiang R, Zhuang Y, Li Y, et al (2025) Mle-dojo: Interactive environments for empowering llm agents in machine learning engineering. URL https://arxiv.org/abs/2505.07782, arXiv:2505.07782

Qiao S, Wei Y, Wang X, et al (2026) Innoeval: On research idea evaluation as a knowledge-grounded, multi-perspective reasoning problem. URL https://arxiv.org/abs/2602.14367, arXiv:2602.14367

Qu A, Zheng H, Zhou Z, et al (2026) Coral: Towards autonomous multi-agent evolution for open-ended discovery. URL https://arxiv.org/abs/2604.01658, arXiv:2604.01658

Qu Y, Lu M (2026) Bilevel autoresearch: Meta-autoresearching itself. URL https://arxiv.org/abs/2603.23420, arXiv:2603.23420

Rasheed RA, Banerjee S, Mukherjee A, et al (2026) From fluent to verifiable: Claim-level auditability for deep research agents. URL https://arxiv.org/abs/2602.13855, arXiv:2602.13855

Rezaei M, Vacareanu R, Wang Z, et al (2025) Online rubrics elicitation from pairwise comparisons. URL https://arxiv.org/abs/2510.07284, arXiv:2510.07284

Ríos-García M, Alampara N, Gupta C, et al (2026) Ai scientists produce results without reasoning scientifically. URL https://arxiv.org/abs/2604.18805, arXiv:2604.18805

Roig J (2025) Towards a standard, enterprise-relevant agentic ai benchmark: Lessons from 5.5 billion tokens' worth of agentic ai evaluations. URL https://arxiv.org/abs/2511.08042, arXiv:2511.08042

Romera-Paredes B, Barekatain M, Novikov A, et al (2024) Mathematical discoveries from program search with large language models. Nature 625(7995):468–475. https://doi.org/10.1038/s41586-023-06924-6, URL https://doi.org/10.1038/s41586-023-06924-6

Roohani Y, Lee A, Huang Q, et al (2024) Biodiscoveryagent: An ai agent for designing genetic perturbation experiments. URL https://arxiv.org/abs/2405.17631, arXiv:2405.17631

Rothfarb S, Davis MC, Matanovic I, et al (2025) Hierarchical multi-agent large language model reasoning for autonomous functional materials discovery. URL https://arxiv.org/abs/2512.13930, arXiv:2512.13930

Roy S, Hegde DV, Schmude J, et al (2025) Suryabench: Benchmark dataset for advancing machine learning in heliophysics and space weather prediction. URL https://arxiv.org/abs/2508.14107, arXiv:2508.14107

Sadhuka S, Prinster D, Fannjiang C, et al (2025) E-valuator: Reliable agent verifiers with sequential hypothesis testing. URL https://arxiv.org/abs/2512.03109, arXiv:2512.03109

Sahoo D, Prasad M, Majhi V, et al (2025) When reject turns into accept: Quantifying the vulnerability of llm-based scientific reviewers to indirect prompt injection. URL https://arxiv.org/abs/2512.10449, arXiv:2512.10449

Salimi M, Mohtasham BH, Aghakasiri A, et al (2026) Debate as reward: A multi-agent reward system for scientific ideation via rl post-training. URL https://arxiv.org/abs/2604.16723, arXiv:2604.16723

Schmidgall S, Moor M (2025) Agentrxiv: Towards collaborative autonomous research. URL https://arxiv.org/abs/2503.18102, arXiv:2503.18102

Schmidgall S, Su Y, Wang Z, et al (2025) Agent laboratory: Using llm agents as research assistants. URL https://arxiv.org/abs/2501.04227, arXiv:2501.04227

Seo M, Baek J, Lee S, et al (2025) Paper2code: Automating code generation from scientific papers in machine learning. URL https://arxiv.org/abs/2504.17192, arXiv:2504.17192

Shao C, Huang D, Li Y, et al (2025) Omniscientist: Toward a co-evolving ecosystem of human and ai scientists. URL https://arxiv.org/abs/2511.16931, arXiv:2511.16931

Sharma M, Zhang CBC, Bandi C, et al (2025) Researchrubrics: A benchmark of prompts and rubrics for evaluating deep research agents. URL https://arxiv.org/abs/2511.07685, arXiv:2511.07685

Shen WF, Qiu X, Whitehouse C, et al (2026a) Rethinking rubric generation for improving llm judge and reward modeling for open-ended tasks. URL https://arxiv.org/abs/2602.05125, arXiv:2602.05125

Shen Y, Yi Z, Zhao Z, et al (2026b) An empirical study of multi-agent collaboration for automated research. URL https://arxiv.org/abs/2603.29632, arXiv:2603.29632

Sheng H, Liu X, He H, et al (2025) Analyzing uncertainty of llm-as-a-judge: Interval evaluations with conformal prediction. URL https://arxiv.org/abs/2509.18658, arXiv:2509.18658

Sherwood J, Aybar B, Kaplan B (2026) Frontier coding agents can now implement an alphazero self-play machine learning pipeline for connect four that performs comparably to an external solver. URL https://arxiv.org/abs/2604.25067, arXiv:2604.25067

Shi L, Zheng ZJ, Juan X, et al (2026a) Qumus: Realization of an embodied ai quantum material experimentalist. URL https://arxiv.org/abs/2605.18407, arXiv:2605.18407

Shi T, Li Y, Wang Z, et al (2026b) Knowledge-driven autonomous materials research via collaborative multi-agent and robotic system. Matter 9(2):102577. https://doi.org/10.1016/j.matt.2025.102577, URL https://doi.org/10.1016/j.matt.2025.102577

Shojaee P, Nguyen NH, Meidani K, et al (2025) Llm-srbench: A new benchmark for scientific equation discovery with large language models. URL https://arxiv.org/abs/2504.10415, arXiv:2504.10415

Si C, Yang D, Hashimoto T (2024) Can llms generate novel research ideas? a large-scale human study with 100+ nlp researchers. URL https://arxiv.org/abs/2409.04109, arXiv:2409.04109

Siegel ZS, Kapoor S, Nagdir N, et al (2024) Core-bench: Fostering the credibility of published research through a computational reproducibility agent benchmark. URL https://arxiv.org/abs/2409.11363, arXiv:2409.11363

Singh V, Ganguly D, Yu H, et al (2026) Toward guarantees for clinical reasoning in vision language models via formal verification. URL https://arxiv.org/abs/2602.24111, arXiv:2602.24111

Skarlinski M, Cox S, Laurent JM, et al (2024) Language agents achieve superhuman synthesis of scientific knowledge. URL https://arxiv.org/abs/2409.13740, arXiv:2409.13740

Son G, Hong J, Fan H, et al (2025) When ai co-scientists fail: Spot-a benchmark for automated verification of scientific research. URL https://arxiv.org/abs/2505.11855, arXiv:2505.11855

Song K, Trotter A, Chen JY (2025) Llm agent swarm for hypothesis-driven drug discovery. URL https://arxiv.org/abs/2504.17967, arXiv:2504.17967

Song X, Chang H, Dong G, et al (2026) Envscaler: Scaling tool-interactive environments for llm agent via programmatic synthesis. URL https://arxiv.org/abs/2601.05808, arXiv:2601.05808

Srivastava G, Hussain AH, Bi Z, et al (2025) Beyondbench: Contamination-resistant evaluation of reasoning in language models. URL https://arxiv.org/abs/2509.24210, arXiv:2509.24210

Starace G, Jaffe O, Sherburn D, et al (2025) Paperbench: Evaluating ai's ability to replicate ai research. URL https://arxiv.org/abs/2504.01848, arXiv:2504.01848

Staufer L, Feng K, Wei K, et al (2026) The 2025 ai agent index: Documenting technical and safety features of deployed agentic ai systems. URL https://arxiv.org/abs/2602.17753, arXiv:2602.17753

Sun Y, Wang H, Li D, et al (2025) The emperor's new clothes in benchmarking? a rigorous examination of mitigation strategies for llm benchmark data contamination. URL https://arxiv.org/abs/2503.16402, arXiv:2503.16402

Sun Y, Ren X, Zhang K, et al (2026a) PerspectiveGap: A benchmark for multi-agent orchestration prompting. URL https://arxiv.org/abs/2606.08878, arXiv:2606.08878

Sun Y, Wen Y, Yang H (2026b) ReSearch: A multi-stage machine learning framework for earth science data discovery. URL https://arxiv.org/abs/2601.14176, arXiv:2601.14176

Tang J, Xia L, Li Z, et al (2025) Ai-researcher: Autonomous scientific innovation. URL https://arxiv.org/abs/2505.18705, arXiv:2505.18705

Tang Y, Yang Y (2026) Ai research agents narrow scientific exploration. URL https://arxiv.org/abs/2605.27905, arXiv:2605.27905

Thakkar N, Yuksekgonul M, Silberg J, et al (2025) Can llm feedback enhance review quality? a randomized study of 20k reviews at iclr 2025. URL https://arxiv.org/abs/2504.09737, arXiv:2504.09737

Theiler R, Comito L, Leko D, et al (2026) From paper to benchmark: agentic, framework-based reproduction of under-specified methods in machine health intelligence. URL https://arxiv.org/abs/2605.28371, arXiv:2605.28371

Thind R, Sun Y, Liang L, et al (2025) OptimAI: Optimization from natural language using llm-powered ai agents. URL https://arxiv.org/abs/2504.16918, arXiv:2504.16918

Tian M, Gao L, Zhang SD, et al (2024) Scicode: A research coding benchmark curated by scientists. URL https://arxiv.org/abs/2407.13168, arXiv:2407.13168

Tie G, Zhou P, Sun L (2025) A survey of ai scientists. URL https://arxiv.org/abs/2510.23045, arXiv:2510.23045

Tie G, Shi J, Song D, et al (2026) Autoresearch ai: Towards ai-powered research automation for scientific discovery. arXiv:2605.23204

Ting YS, Curtis-Trudel A, Yao S (2026) What understanding means in ai-laden astronomy. URL https://arxiv.org/abs/2601.10038, arXiv:2601.10038

Toledo E, Hambardzumyan K, Josifoski M, et al (2025) Ai research agents for machine learning: Search, exploration, and generalization in mle-bench. URL https://arxiv.org/abs/2507.02554, arXiv:2507.02554

Trirat P, Jeong W, Hwang S (2024) Automl-agent: A multi-agent llm framework for full-pipeline automl. URL https://arxiv.org/abs/2410.02958, arXiv:2410.02958

Tytler K (2025) Adoption, usability and perceived clinical value of a uk ai clinical reference platform (iatrox): a mixed-methods formative evaluation of real-world usage and a 1,223-respondent user survey. URL https://arxiv.org/abs/2509.21188, arXiv:2509.21188

Wan S, Nikolaidis C, Song D, et al (2024) Cyberseceval 3: Advancing the evaluation of cybersecurity risks and capabilities in large language models. URL https://arxiv.org/abs/2408.01605, arXiv:2408.01605

Wang L, He Y, Chen P, et al (2026a) Time to reflect: Can we trust llm judges for evidence-based research agents? URL https://arxiv.org/abs/2605.19196, arXiv:2605.19196

Wang S, Tian X, Liu P, et al (2026b) Workflow closure is not scientific closure in auto-research systems. URL https://arxiv.org/abs/2605.26200, arXiv:2605.26200

Wang X, Wang J, Wang Y, et al (2025a) Mars: toward more efficient multi-agent collaboration for llm reasoning. URL https://arxiv.org/abs/2509.20502, arXiv:2509.20502

Wang Y, Luan Z (2026) Parness: A paper harness for end-to-end automated scientific research with dynamic workflows, full-text indexing, and cross-run knowledge accumulation. URL https://arxiv.org/abs/2605.05258, arXiv:2605.05258

Wang Y, Qian Y, Zhang Y, et al (2026c) Deploybench: Benchmarking llm agents for research artifact deployment. URL https://arxiv.org/abs/2606.05238, arXiv:2606.05238

Wang Y, Sheng R, Shao E, et al (2026d) Figures as interfaces: Toward llm-native artifacts for scientific discovery. URL https://arxiv.org/abs/2604.08491, arXiv:2604.08491

Wang Y, Wang L, Deng Y, et al (2026e) Deepresearcheval: An automated framework for deep research task construction and agentic evaluation. URL https://arxiv.org/abs/2601.09688, arXiv:2601.09688

Wang Y, Zhang Z, Chi M, et al (2026f) Evomembench: Benchmarking agent memory from a self-evolving perspective. URL https://arxiv.org/abs/2605.18421, arXiv:2605.18421

Wang Z, Jung J, Lu X, et al (2025b) Profbench: Multi-domain rubrics requiring professional knowledge to answer and judge. URL https://arxiv.org/abs/2510.18941, arXiv:2510.18941

Wang Z, Siu V, Ye Z, et al (2025c) Agentvigil: Generic black-box red-teaming for indirect prompt injection against llm agents. URL https://arxiv.org/abs/2505.05849, arXiv:2505.05849

Wang Z, Bai F, Luo Z, et al (2026g) Fire-bench: Evaluating agents on the rediscovery of scientific insights. URL https://arxiv.org/abs/2602.02905, arXiv:2602.02905

Wei H, He S, Xia T, et al (2024) Systematic evaluation of llm-as-a-judge in llm alignment tasks: Explainable metrics and diverse prompt templates. URL https://arxiv.org/abs/2408.13006, arXiv:2408.13006

Wei J, Sun Z, Papay S, et al (2025a) Browsecomp: A simple yet challenging benchmark for browsing agents. URL https://arxiv.org/abs/2504.12516, arXiv:2504.12516

Wei J, Yang Y, Zhang X, et al (2025b) From ai for science to agentic science: A survey on autonomous scientific discovery. arXiv:2508.14111

Wei T, Sachdeva N, Coleman B, et al (2025c) Evo-memory: Benchmarking llm agent test-time learning with self-evolving memory. URL https://arxiv.org/abs/2511.20857, arXiv:2511.20857

Weidener L, Brkić M, Jovanović M, et al (2026a) Rethinking the ai scientist: Interactive multi-agent workflows for scientific discovery. URL https://arxiv.org/abs/2601.12542, arXiv:2601.12542

Weidener L, Brkić M, Lee P, et al (2026b) From agent-only social networks to autonomous scientific research: Lessons from openclaw and moltbook, and the architecture of clawdlab and beach.science. URL https://arxiv.org/abs/2602.19810, arXiv:2602.19810

Weng Y, Zhu M, Bao G, et al (2024) Cycleresearcher: Improving automated research via automated review. URL https://arxiv.org/abs/2411.00816, arXiv:2411.00816

Weng Y, Zhu M, Xie Q, et al (2025) Deepscientist: Advancing frontier-pushing scientific findings progressively. URL https://arxiv.org/abs/2509.26603, arXiv:2509.26603

Weng Y, Zhu M, Xie Q, et al (2026) Deepreviewer 2.0: A traceable agentic system for auditable scientific peer review. URL https://arxiv.org/abs/2604.09590, arXiv:2604.09590

Wijk H, Lin T, Becker J, et al (2024) Re-bench: Evaluating frontier ai r&d capabilities of language model agents against human experts. URL https://arxiv.org/abs/2411.15114, arXiv:2411.15114

Wu B, Wang S, Liu Y, et al (2026a) From automated to autonomous: Hierarchical agent-native network architecture (hana). URL https://arxiv.org/abs/2605.20608, arXiv:2605.20608

Wu H, Zheng B, Song D, et al (2026b) Towards a medical ai scientist. URL https://arxiv.org/abs/2603.28589, arXiv:2603.28589

Wu M, Chen P, Hong G, et al (2025) Mcpzoo: A large-scale dataset of runnable model context protocol servers for ai agent. URL https://arxiv.org/abs/2512.15144, arXiv:2512.15144

Wu S, Choi E, Khatua A, et al (2026c) Humanlm: Simulating users with state alignment beats response imitation. URL https://arxiv.org/abs/2603.03303, arXiv:2603.03303

Xi S, Rao V, Payan J, et al (2025) Flaws: A benchmark for error identification and localization in scientific papers. URL https://arxiv.org/abs/2511.21843, arXiv:2511.21843

Xiang Y, Yan H, Ouyang S, et al (2025) Scireplicate-bench: Benchmarking llms in agent-driven algorithmic reproduction from research papers. URL https://arxiv.org/abs/2504.00255, arXiv:2504.00255

Xie Q, Xiong Q, Zhu H, et al (2026a) Dr^3-eval: Towards realistic and reproducible deep research evaluation. URL https://arxiv.org/abs/2604.14683, arXiv:2604.14683

Xie S, Shi Z, Shen H, et al (2026b) Deepweb-bench: A deep research benchmark demanding massive cross-source evidence and long-horizon derivation. URL https://arxiv.org/abs/2605.21482, arXiv:2605.21482

Xiong L, Luo K, Xia Z, et al (2026) Autoresearchbench: Benchmarking ai agents on complex scientific literature discovery. URL https://arxiv.org/abs/2604.25256, arXiv:2604.25256

Xu J, Zhu Q, Wu Y, et al (2026) Nanoresearch: Co-evolving skills, memory, and policy for personalized research automation. URL https://arxiv.org/abs/2605.10813, arXiv:2605.10813

Xu L, Borrett T (2026) Beyond ai as assistants: Toward autonomous discovery in cosmology. URL https://arxiv.org/abs/2605.14791, arXiv:2605.14791

Xu R, Chen J, Ye J, et al (2025a) Incentivizing agentic reasoning in llm judges via tool-integrated reinforcement learning. URL https://arxiv.org/abs/2510.23038, arXiv:2510.23038

Xu W, Liang Z, Mei K, et al (2025b) A-mem: Agentic memory for llm agents. URL https://arxiv.org/abs/2502.12110, arXiv:2502.12110

Yamada Y, Lange R, Lu C, et al (2025) The ai scientist-v2: Workshop-level automated scientific discovery via agentic tree search. URL https://arxiv.org/abs/2504.08066, arXiv:2504.08066

Yang J, Jimenez CE, Wettig A, et al (2024) Swe-agent: Agent-computer interfaces enable automated software engineering. URL https://arxiv.org/abs/2405.15793, arXiv:2405.15793

Yang R, Chen R, Kelaita P, et al (2026a) Judgmentbench: Comparing rubric and preference evaluation for quality assessment. URL https://arxiv.org/abs/2605.25240, arXiv:2605.25240

Yang R, Li Y, Li S (2026b) Aris: Autonomous research via adversarial multi-agent collaboration. URL https://arxiv.org/abs/2605.03042, arXiv:2605.03042

Yang X, Yang X, Fang S, et al (2025a) R&d-agent: An llm-agent framework towards autonomous data science. URL https://arxiv.org/abs/2505.14738, arXiv:2505.14738

Yang Z, Wan Y, Yan S, et al (2025b) Drugmcts: a drug repurposing framework combining multi-agent, rag and monte carlo tree search. URL https://arxiv.org/abs/2507.07426, arXiv:2507.07426

Yang Z, Liu X, Xu X (2026c) Sciintegrity-bench: A benchmark for evaluating academic integrity in ai scientist systems. URL https://arxiv.org/abs/2605.10246, arXiv:2605.10246

Ye R, Pang X, Chai J, et al (2024) Are we there yet? revealing the risks of utilizing large language models in scholarly peer review. URL https://arxiv.org/abs/2412.01708, arXiv:2412.01708

Yu G, Wang X (2026) Knows: Agent-native structured research representations. URL https://arxiv.org/abs/2604.17309, arXiv:2604.17309

Zeng Z, Liu J, Chen S, et al (2025) Futurex: An advanced live benchmark for llm agents in future prediction. URL https://arxiv.org/abs/2508.11987, arXiv:2508.11987

Zhai Y, Tao S, Chen C, et al (2025) Agentevolver: Towards efficient self-evolving agent system. URL https://arxiv.org/abs/2511.10395, arXiv:2511.10395

Zhang AK, Perry N, Dulepet R, et al (2024) Cybench: A framework for evaluating cybersecurity capabilities and risks of language models. URL https://arxiv.org/abs/2408.08926, arXiv:2408.08926

Zhang AK, Ji J, Menders C, et al (2025a) Bountybench: Dollar impact of ai agent attackers and defenders on real-world cybersecurity systems. URL https://arxiv.org/abs/2505.15216, arXiv:2505.15216

Zhang B, Feng S, Yan X, et al (2025b) Novelseek: When agent becomes the scientist - building closed-loop system from hypothesis to verification. URL https://arxiv.org/abs/2505.16938, arXiv:2505.16938

Zhang C, Yakobson BI (2026) Matclaw: An autonomous code-first llm agent for end-to-end materials exploration. URL https://arxiv.org/abs/2604.02688, arXiv:2604.02688

Zhang C, Park Y, Fleischer F, et al (2026a) Sok: Darpa's ai cyber challenge (aixcc): Competition design, architectures, and lessons learned. URL https://arxiv.org/abs/2602.07666, arXiv:2602.07666

Zhang J, Fu Z, Xi Z, et al (2026b) Agentv-rl: Scaling reward modeling with agentic verifier. URL https://arxiv.org/abs/2604.16004, arXiv:2604.16004

Zhang J, Lao J, Liu C, et al (2026c) Vfeagent: A multimodal agent framework for end-to-end automated finite element analysis. URL https://arxiv.org/abs/2605.28978, arXiv:2605.28978

Zhang K, Wang X, Zhao H, et al (2026d) Tianji:an autonomous ai meteorologist for discovering physical mechanisms in atmospheric science. URL https://arxiv.org/abs/2603.27738, arXiv:2603.27738

Zhang M, Shen T, Luo H, et al (2026e) Skillflow: Flow-driven recursive skill evolution for agentic orchestration. URL https://arxiv.org/abs/2605.14089, arXiv:2605.14089

Zhang Q, Zhou J, Wang Y, et al (2026f) Rubricbench: Aligning model-generated rubrics with human standards. URL https://arxiv.org/abs/2603.01562, arXiv:2603.01562

Zhang T, Liu Z, Xin Y, et al (2025c) Mooseagent: A llm based multi-agent framework for automating moose simulation. URL https://arxiv.org/abs/2504.08621, arXiv:2504.08621

Zhang TJ, Dev G, Wang N, et al (2025d) Test of time: Rethinking temporal signal of benchmark contamination. URL https://arxiv.org/abs/2509.00072, arXiv:2509.00072

Zhang W, Li X, Zhang Y, et al (2025e) Deep research: A survey of autonomous research agents. arXiv:2508.12752

Zhang Y, Diddee H, Holm S, et al (2025f) Noveltybench: Evaluating language models for humanlike diversity. URL https://arxiv.org/abs/2504.05228, arXiv:2504.05228

Zhang Y, Khalifa M, Bhushan S, et al (2025g) Mlrc-bench: Can language agents solve machine learning research challenges? URL https://arxiv.org/abs/2504.09702, arXiv:2504.09702

Zhang Y, Cheng X, Liu T, et al (2026g) Drugsage: Self-evolving agent experience for efficient state-of-the-art drug discovery. URL https://arxiv.org/abs/2605.15461, arXiv:2605.15461

Zhang Z, Wang N, Galhotra S, et al (2026h) How far are we from true auto-research? URL https://arxiv.org/abs/2605.19156, arXiv:2605.19156

Zhang Z, Yin A, Baweja A, et al (2026i) El agente forjador: Task-driven agent generation for quantum simulation. URL https://arxiv.org/abs/2604.14609, arXiv:2604.14609

Zhao H, Zhu Y, Wang Z, et al (2025a) Confagents: A conformal-guided multi-agent framework for cost-efficient medical diagnosis. URL https://arxiv.org/abs/2508.04915, arXiv:2508.04915

Zhao K, Xu F, Li Y, et al (2026) Hybridquestion: Human-ai collaboration for identifying high-impact research questions. URL https://arxiv.org/abs/2602.03849, arXiv:2602.03849

Zhao X, Sang Z, Li Y, et al (2025b) Autoreproduce: Automatic ai experiment reproduction with paper lineage. URL https://arxiv.org/abs/2505.20662, arXiv:2505.20662

Zhou B, Huang X, Ning H, et al (2026a) Nora: A harness-engineered autonomous research agent for end-to-end spatial data science. URL https://arxiv.org/abs/2605.02092, arXiv:2605.02092

Zhou K, Woo S, Ding H, et al (2026b) An empirical study of automating agent evaluation. URL https://arxiv.org/abs/2605.11378, arXiv:2605.11378

Zhou M, Yao Q, Du L, et al (2025) Reflective paper-to-code reproduction enabled by fine-grained verification. URL https://arxiv.org/abs/2508.16671, arXiv:2508.16671

Zhou Z, Tang H, Zhang J, et al (2026c) Can current agents close the discovery-to-application gap? a case study in minecraft. URL https://arxiv.org/abs/2604.24697, arXiv:2604.24697

Zhu B, Jia Q, Lan T, et al (2026a) Marco deepresearch: Unlocking efficient deep research agents via verification-centric design. URL https://arxiv.org/abs/2603.28376, arXiv:2603.28376

Zhu C, Xiong J, Ma R, et al (2025a) When your reviewer is an llm: Biases, divergence, and prompt injection risks in peer review. URL https://arxiv.org/abs/2509.09912, arXiv:2509.09912

Zhu M, Tan J, Murthy R, et al (2026b) Realusersim: Bridging the reality gap in agent benchmarking via grounded user simulation. URL https://arxiv.org/abs/2605.20204, arXiv:2605.20204

Zhu X, Cai Y, Liu Z, et al (2026c) Toward ultra-long-horizon agentic science: Cognitive accumulation for machine learning engineering. URL https://arxiv.org/abs/2601.10402, arXiv:2601.10402

Zhu Z, Lin KQ, Shou MZ (2025b) Paper2video: Automatic video generation from scientific papers. URL https://arxiv.org/abs/2510.05096, arXiv:2510.05096

Zhuge M, Zhao C, Ashley DR, et al (2024) Agent-as-a-judge: Evaluate agents with agents. URL https://arxiv.org/abs/2410.10934, arXiv:2410.10934

Zou Q, Lam HH, Zhao W, et al (2025) Fml-bench: Benchmarking machine learning agents for scientific research. URL https://arxiv.org/abs/2510.10472, arXiv:2510.10472

Zou Q, Lam HH, Zhao W, et al (2026) Fml-bench: A controlled study of ai research agent strategies from the perspective of search dynamics. URL https://arxiv.org/abs/2605.17373, arXiv:2605.17373

Zubiaga I, Soroa A, Agerri R (2026) Towards reliable multilingual llms-as-a-judge: An empirical study. URL https://arxiv.org/abs/2605.28710, arXiv:2605.28710