

Agent Systems for Academic Research Automation

An Evolving Survey

Pierfrancesco Beneventano^{*1}, Riccardo Neumarker^{*1},
Mahmoud Abdelmoneum¹, Marc Bacvanski¹, Yulu Gan¹, Mehdi Hajoub¹,
Qianli Liao¹, Emanuele Rimoldi¹, Kushagra Tiwary¹, Liu Ziyin^{1,2},
Tomer Galanti³, Theodoros Evgeniou⁴, Tomaso Poggio¹

¹*Massachusetts Institute of Technology* ²*NTT Research*
³*Texas A&M University* ⁴*INSEAD*

V1: April 2, 2026, Current: April 8, 2026

Abstract. Agentic systems are beginning to reshape scholarly work, e.g., they can now design experiments, develop mathematical arguments, generate full academic papers, peer reviews, and rebuttals. This raises a natural question: to what extent can research itself be automated? The recent history of research automation can be read as a progression from retrieval and indexing, to citation linking, to summarization, and finally to end-to-end agentic systems. This survey examines the agentic systems most directly relevant to that question, namely those whose primary outputs are scholarly artifacts. First, we distill the recurring engineering principles and architectural patterns in current systems. Second, we propose a compact conceptual framework for identifying how to map the systems within the landscape as the field evolves. For the latter, we organize these systems along three dimensions: the phases of the research process a system covers, the kind of artifacts it produces, and the verification regime that governs its principal claims. Because this area is evolving faster than traditional publication cycles can accommodate, we treat the survey as a living document, updated regularly to track one of the fastest-moving developments in contemporary research automation.

*Equal contribution. Correspondence: pierb@mit.edu. Please check [pAI website](#) for updates. Please contact us for any inconsistencies, errors, to join our efforts, or to include your work in this survey.

1 Introduction

The term “research agent” is now used to describe a rapidly growing and highly heterogeneous set of systems. Under that single label sits citation-grounded section writers, literature-review generators, full-paper pipelines, review and rebuttal agents, manuscript-state revision tools, and broader scholarly copilots. These systems are not minor variants of a single system architecture, but are technically different solutions for different scholarly tasks, operating under different measures of quality and requirements for evidence.

The scope. While automating academic research has been an overarching goal for a few decades [9, 40, 70], recent progress has been accelerated by reasoning-oriented models [27, 49] and end-to-end agentic research systems [1, 45, 46, 58]. However, we believe that the familiar distinctions between, e.g., reasoning models, single-agent, multi-agent systems, is the wrong point of view to explain the current landscape. A reviewer-panel system, a planner-executor manuscript pipeline, and a project-state writing assistant should not be treated as equivalent merely because each distributes work across multiple roles. Indeed, two systems may both be multi-agent while differing fundamentally in what they produce (artifact type), which stages of the research process and what field they cover (research task), how coordination is organized (architecture), and what mechanisms can falsify, constrain, or block erroneous output (verifier/evaluation).

This survey adopts a deliberately manuscript-centered perspective. We focus on agentic systems whose primary outputs are scholarly artifacts, ranging from individual sections and literature reviews to complete papers, peer reviews, rebuttals, and structured revisions. The main questions we answer are the following two. First, we ask **Question 1**:

What engineering principles recur across the strongest current systems? (1)

Secondly, we address **Question 2**:

What conceptual framework allows a newly introduced system to be mapped in the landscape as the field continues to evolve? (2)

To answer the first question, we distill the design principles and architectural patterns that repeatedly emerge across state-of-the-art systems. To answer the second, we organize the space along three axes: which phases of what research lifecycle a system covers, the artifacts it produces, and the verification regime that governs its principal claims.

From an historical perspective, research automation was a progression from retrieval and indexing, to citation linking, to summarization, and finally, recently, to end-to-end agentic systems. Across all of these stages, one constraint recurs: *generated scholarly text must remain tied to external evidence strong enough to justify its claims*. The central question for a research agent is therefore not how autonomous it appears, but whether the scope of its manuscript claims is matched to the strongest verifier the architecture can actually bring to bear. When that match breaks, the result is not merely hallucination in the narrow sense, but a document that presents itself as scientifically resolved before the underlying science has in fact been adequately resolved. Local quality is not global manuscript validity. Later in the paper, we analyze this document-level failure as *closure failure*. Accordingly, the way we contribute with this manuscript is by

- Defining a clean comparison set for such systems,
- Proposing a comparison framework for the new ones,
- Arguing in Section 7 for a different theory of failure and evaluation than the one the field currently uses.

Because this area is moving substantially faster than traditional publication cycles, we treat the survey as a living document rather than a static retrospective. New systems appear continuously, often with uneven documentation, shifting terminology, and changing empirical support. A useful survey must therefore do more than catalogue examples: it must provide a stable conceptual map that remains informative even as the underlying landscape changes.

Organization. The remainder of the paper is organized as follows. Section 2 defines the manuscript-centered scope of inclusion and the evidence tiers used throughout. Section 3 distills the engineering principles that recur across current systems architectures, thus responding to **Question 1** above. The following sections address **Question 2**. Section 4 introduces the paper’s prospective map, locating systems by task coverage, artifact type, and verification regime. Section 5 then surveys the main system families in practice, corresponding to recurrent artifact types in the current literature. Section 6 examines how those families change across different verification regimes, from formally verifiable symbolic domains to open empirical biological and clinical ones. Section 7 analyzes the fragmented evaluation landscape, and Section 8 turns to structural limits, epistemic ownership, and institutional choice.

Contents

1	Introduction	2
2	Scope and Evidence Tiers	4
3	Fundamental Principles of Engineering Research Agents	5
3.1	Systems must be designed around manuscript obligations, not agent count.	5
3.2	Claims Require Matched Grounding and Matched Verifiers	6
3.3	Coordination Should Follow Real Bottlenecks	6
3.4	Manuscript production requires explicit project state, traceability, and revision memory.	7
3.5	Critique Improves Text but Does Not Verify Claims	7
3.6	Local Validity Does Not Close the Manuscript	8
3.7	Autonomy Is Limited by Verifier Strength, Cost, and Latency	8
4	A Taxonomy of Research Agent Systems	9
4.1	Task Coverage as Positional Description	9
4.2	Artifact Type	10
4.3	Verification Regime	10
5	Systems in Practice	11
5.1	Family I: Section-Level Evidence-Grounded Writing	11
5.2	Family II: Survey and Literature-Review Generation	12
5.3	Family III: Full-Paper and End-to-End Research Agents	13

5.4	Family IV: Review, Revision, and Rebuttal Agents	14
6	Domain-Specific Applications	15
6.1	Formally Verifiable Symbolic Domains	16
6.2	Executable Code-and-Data Domains	17
6.3	Tool-Mediated Physical Sciences	18
6.4	Open Empirical Biological and Clinical Domains	19
7	Benchmarking and Evaluation	20
7.1	Section-Level Grounding and Survey Synthesis	21
7.2	Full-Paper Pipeline Quality	21
7.3	Review and Revision Utility	22
7.4	The Unresolved Object of Evaluation	23
8	Structural Limits, Epistemic Ownership, and Institutional Choice	23
9	Conclusion	26
A	Closed Products and Adjacent Copilots	36

2 Scope and Evidence Tiers

This survey adopts a strict manuscript-centered criterion for inclusion. We include systems whose primary function is to generate or revise manuscript-like artifacts: citation-grounded sections, literature reviews, full papers, review or rebuttal text, or structured edits over a live manuscript state. To support meaningful comparison, we require enough technical documentation to determine which phases of the research lifecycle a system covers, what scholarly artifact it primarily produces, and what verification regime governs its core claims. Systems whose primary output is instead a code change, a benchmark improvement, or an experimental result without manuscript production as a first-class objective are excluded from the core comparison set, although they may appear as contextual examples where helpful.

This definition is intentionally narrower than the broad categories of “AI for science” or “research automation.” A system that automates laboratory protocols, runs closed-loop experiments, or produces benchmark results may be scientifically important while still being a different kind of object from a system whose main role is to write, revise, or critique the paper about those results. Conflating these categories obscures the design pressures relevant to each and weakens comparison on both sides.

Boundary cases. Several systems lie near the boundary of our scope and therefore require explicit treatment. FARS by Analemma publicly presents itself as an end-to-end system spanning ideation, experimentation, and paper writing, which makes it relevant to this survey; we therefore include it as a boundary case, while assigning lower evidentiary weight because the public technical documentation remains limited [2, 3]. AUTORESEARCH is better understood as an autonomous ML experimentation loop whose main artifact is an iteratively modified training script together with measured performance changes, rather than a manuscript [38, 39]. AUTORESEARCH@HOME moves somewhat closer to a scientific workflow by adding distributed coordination and public reporting, but manuscript production is still secondary to the experimental loop itself [5, 18, 48]. It is therefore excluded from the core comparison set.

Evidence tiers. Each system in our analysis is assigned an evidence tier that qualifies the strength of the claims we make about its design and performance. The tiers range from peer-reviewed archival publications (T1), to preprints accompanied by technical artifacts (T2), to official documentation and repositories (T3), and finally to informal announcements or demonstrations without reproducible technical detail (T4). This distinction is not merely administrative. A system may be highly visible or influential while still being documented only at T3 or T4; this does not make it unimportant, but it does limit what can be responsibly claimed about its architecture, empirical behavior, and validation mechanism. Throughout the survey, we therefore treat public visibility and evidentiary reliability as distinct properties.

3 Fundamental Principles of Engineering Research Agents

The systems surveyed here do not improve simply by adding more agents, more tools, or longer context windows. What recurs instead is a smaller set of engineering constraints, imposed by the kind of scholarly artifact the system must produce and the verification conditions under which it must produce it. A section writer, a survey generator, a full-paper pipeline, and a review agent face different constraints because they are answerable to different objects: different evidence requirements, different correctness criteria, and different points where errors become consequential. This section reconstructs those constraints from the systems, evaluation landscapes, and failure modes examined in the sections that follow, and distills them into a set of recurring principles.

3.1 Systems must be designed around manuscript obligations, not agent count.

The primary design object in this survey is the scholarly artifact for which the system is answerable. A section-writing system such as SCHOLARCOPILOT is evaluated mainly by claim–citation alignment at the paragraph level [72]. Survey systems such as AUTOSURVEY, SURVEYFORGE, SURVEYX, SURVEYGEN-I, ARISE, OPENSCHOLAR, and PAPERQA2 are evaluated instead by the quality of corpus selection, synthesis, contradiction handling, and long-range document coherence [4, 12, 43, 62, 71, 73, 82]. Full-paper systems do not occupy a single stage profile. Systems such as DATA-TO-PAPER, AI-RESEARCHER, AGENT LABORATORY, and THE AI SCIENTIST shift the center of gravity toward stages 2–7, because the manuscript is expected to package an underlying research process rather than retrieved literature alone [32, 45, 46, 58, 67, 81]. By contrast, systems such as PAPERORCHESTRA are concentrated later in the cycle, around manuscript drafting and packaging, because they assume completed research materials and treat full-manuscript synthesis itself as the primary task [63]. Review and rebuttal systems such as AGENTREVIEW, REVIEWAGENTS, and SWIF²T are judged not by direct scientific discovery, but by the quality, specificity, and usefulness of the critique they produce [10, 22, 37].

What varies across these families is not merely output length, but the contract the architecture must satisfy: what counts as admissible evidence, which errors are consequential, and where correction must occur. This is why agent count is an analytically weak descriptor. A reviewer panel, an outline-driven survey pipeline, and an experiment-writing loop may all be multi-agent while addressing different scholarly objects under different correctness criteria. The right starting point for a system design is therefore the manuscript obligation itself. Architectures built around generic notions of research capability often optimize intermediate proxies such as fluency, local factuality,

or task completion rate, even when those quantities are secondary to the scientific reliability of the final scholarly artifact.

3.2 Claims Require Matched Grounding and Matched Verifiers

Across the systems considered here, no single grounding strategy—that is, no single way of tying manuscript claims to supporting evidence or executable state—is sufficient. Some manuscript claims are primarily literature-backed and can be constrained through retrieval, citation selection, and support checking, which explains the central role of retrieval-conditioned generation in systems such as SCHOLARCOPILOT, OPENSCHOLAR, and PAPERQA2 [4, 62, 72]. Retrieval alone, however, is too weak for empirical performance claims. In executable domains, the architecture must tie writing to code execution, rerunnable analysis, and provenance tracking, as in DATA-TO-PAPER, AI-RESEARCHER, AGENT LABORATORY, and THE AI SCIENTIST [32, 45, 46, 58, 67, 81]. In formally verifiable domains, systems such as ALPHAPROOF and APOLLO go further still: the proof assistant is not merely a tool attached to the loop, but the environment within which meaningful progress occurs [19, 51]. In tool-mediated and open empirical regimes, by contrast, the verifier is weaker, slower, or external to the system, and the architecture must be adjusted accordingly [15, 26, 28, 31, 56, 74].

The principle that emerges is therefore stronger than the idea that grounding and verification are merely two separate design goals. What matters is whether the system can provide a way of checking claims that matches the kind of claim the manuscript makes. Section 7 identifies the same problem from the evaluation side as a mismatch between claim type and validator. Architecturally, that mismatch yields a characteristic failure: the document makes stronger claims than its checking mechanism can justify. The most serious errors in manuscript agents are rarely due to missing text generation alone. They arise when manuscript claims outrun the strongest verifier the architecture can actually bring to bear.

3.3 Coordination Should Follow Real Bottlenecks

The survey does not support a simple single-agent versus multi-agent distinction as a useful analytical axis. Multi-agent design is widespread, but successful decompositions are not defined by the number of roles introduced. They are defined by whether the split occurs at a genuine bottleneck in the research process. In survey systems, that bottleneck may lie between corpus construction, outline formation, and long-form drafting [12, 43, 73, 82]. In full-paper systems, it may lie between conceptual analysis, code mapping, execution, and reporting, as in AI-RESEARCHER, or between exploration and experiment management, as in THE AI SCIENTIST-V2 [46, 67, 81]. In review systems, the split may separate reviewer, author, and area-chair functions, or distinguish local evidence gathering from final recommendation [14, 22, 37, 76, 86].

The analogy to human research organizations is therefore useful only in a limited sense, and it should remain limited. It helps insofar as both humans and agents benefit from division of labor at points where information, judgment, or checking must be handed off from one stage to another. It becomes misleading when those role labels are treated as if they were, by themselves, a theory of good system design. The operative question is not whether the system resembles a social organization, but whether its decomposition preserves what later stages need in order to integrate

results. When it does not, adding agents merely multiplies coordination surfaces, conflicting assumptions, and points of integration failure. Coordination should therefore be understood as a response to real bottlenecks in the workflow, not as a proxy for capability in its own right.

3.4 Manuscript production requires explicit project state, traceability, and revision memory.

Long-horizon manuscript production requires more than a large context window. Survey generation, full-paper writing, and revision all depend on information that must remain available, revisable, and contestable across many steps: outlines, source selections, methodological decisions, experimental outcomes, claim caveats, reviewer comments, and document edits. Across the systems discussed here, the recurring response to that pressure is to maintain explicit project state. SURVEYX uses an AttributeTree to distill and organize source material before generation; SURVEYGEN-I combines adaptive planning with memory-guided writing; SURVEYFORGE adds memory-driven retrieval; DATA-TO-PAPER treats manuscript claims as traceable endpoints of an execution history; and PAPERDEBUGGER tracks live document state and revision history as structured objects rather than as flat text. [12, 30, 32, 43, 82].

This principle concerns both capability and accountability. Manuscripts evolve: related work expands, limitations surface late, experiments fail, reviewers expose weaknesses, and claims must sometimes be narrowed or withdrawn. Without explicit state, continuity can only be approximated through repeated re-prompting, which is precisely where cross-section drift and forgotten constraints emerge. With explicit state, revisions can propagate across the document and leave a clearer record of how a claim entered the paper. In manuscript agents, persistent state is therefore not a convenience feature. It is the substrate on which both long-horizon coherence and later provenance depend.

3.5 Critique Improves Text but Does Not Verify Claims

One of the clearest lessons of review, revision, and rebuttal systems is that critique and verification must remain distinct. Systems such as AGENTREVIEW, REVIEWAGENTS, MARG, DEEPREVIEW, CYCLERESEARCHER/CYCLEREVIEWER, and SWIF²T show that role-conditioned review loops can improve specificity, coverage, actionability, and reviewer-style reasoning [10, 14, 22, 37, 76, 86]. Similar critique loops also appear outside Family IV, for example in ARISE's rubric-guided revision cycles and in reviewer ensembles used inside end-to-end systems [45, 46, 73, 81].

Critique, however, operates mainly within the system's representational space: the draft text, the retrieved sources, the rubric, or the internal deliberation trace. This limitation is not removed simply by using a better model. A critique can identify that a claim seems unsupported, poorly framed, or inconsistent with another section, but it cannot by itself establish whether the underlying experiment was correctly executed, whether a cited source really supports the claim being made, or whether the statistical analysis is valid. Those checks require contact with something external to the text: data, code, formal proofs, or experimental outcomes. This is why the evaluation of review agents is dominated by usefulness, plausibility, and agreement with human review behavior rather than by demonstrated correction of scientific error. When critique is mistaken for verification, the characteristic failure is self-confirming criticism: the prose becomes more polished while the truth conditions of the manuscript's central claims remain unchanged.

3.6 Local Validity Does Not Close the Manuscript

The most stable parts of this survey point to a more fundamental limit: manuscript validity is global, whereas most current validation mechanisms are local. A section may be citation-grounded, a computation may run, a proof fragment may be repaired, and a paragraph may receive useful feedback, while the document as a whole still overstates what has actually been established. Section 7 makes the evaluative version of this point explicit: current benchmarks do not adequately test global manuscript coherence or contradiction handling across sources. Section 8 names the corresponding document-level failure *closure failure*: the manuscript claims scientific certainty before the underlying science conclusively demonstrates it.

This principle fills a major gap in the current conceptual skeleton. It helps explain why survey systems that are locally fluent still struggle with disagreement representation and critical analysis [6]. It explains why manuscript-state tools can improve local sections without guaranteeing repair of the paper's global thesis [30]. It also explains why execution-backed systems still require strong claim discipline: DATA-TO-PAPER addresses the problem by tightening traceability, whereas the THE AI SCIENTIST lineage illustrates how a rhetorically complete paper may still depend on the depth and honesty of the underlying validation [32, 45, 46, 81]. The engineering implication is not merely that systems need more memory, more critique, or even larger context windows. It is that future systems will require richer document representations in which claims, evidential status, uncertainty, and revision dependencies can be propagated across the manuscript as a whole rather than repaired one span at a time. A larger context window that holds the full document does not by itself provide that structure: it makes the text visible, but not the dependency relations that determine whether a revision to one claim should trigger a correction elsewhere.

3.7 Autonomy Is Limited by Verifier Strength, Cost, and Latency

Once the preceding principles are in view, the question of how much autonomy a system can safely exercise becomes more precise. Safe autonomy does not scale with rhetorical fluency or with the number of coordinated agents. It scales with the strength, position, and practical cost of the verifier. In formally verifiable domains, once formalization succeeds, the loop can often be closed internally, allowing substantial autonomy to shift into proof search and repair [19, 51]. In executable code-and-data domains, autonomy can extend further because the system can rerun analyses and compare outcomes, although human oversight remains valuable for framing, novelty, and error interpretation [32, 45, 46, 58, 81]. In tool-mediated physical sciences, local closure around simulators or instruments is already weaker, because tool success is only a proxy for the final scientific claim [15, 28, 66]. In open empirical biology and medicine, where decisive validation remains external, slow, and expensive, the autonomous role is narrower: triaging hypotheses, designing ranked experiments, analyzing existing data, and writing with explicit uncertainty rather than implying that validation has already been completed [26, 31, 56, 74].

The practical conclusion is that autonomy and verifier strength must be treated as coupled design variables. Attempts to maximize one while holding the other fixed reliably produce the failure modes catalogued later in the paper: proxy collapse in tool-mediated domains, speculative manuscript inflation in open empirical settings, and, more generally, the conversion of partial evidence into prose that reads as settled science.

Taken together, these principles recast research-agent engineering not as the construction of generic autonomous writers, but as the design of systems whose outputs can be inspected, contested, and revised along the path from claim to evidence. The strongest current architectures are those that align coordination with real bottlenecks, maintain explicit project state, attach claims to the strongest available verifier, and remain honest about where closure has not yet been achieved. The unresolved frontier is making the resulting manuscript a more faithful representation of what the pipeline has actually established, not merely extending how much of the pipeline the agent covers.

4 A Taxonomy of Research Agent Systems

Section 3 argued that agent count is a weak descriptor of research automation. What is still needed, however, is a compact way to position a newly emerging system before it is absorbed into a retrospective catalog. The taxonomy developed in this survey is therefore three-dimensional. The first dimension is task coverage, understood as the portion of the research lifecycle in which a system's manuscript obligations are concentrated. The second dimension is artifact type, understood as the primary scholarly object for which the system is answerable. The third dimension is verification regime, understood as the bottleneck verifier of the manuscript's principal claims. No single dimension is sufficient on its own. Taken together, they explain why superficially similar systems may require different architectures, and why systems that look different on the surface may nonetheless face the same design pressures.

Research lifecycle map. For orientation, Figure 1 should be read not as a ladder of increasing autonomy, but as a coordinate map. It is used here to position manuscript-centered systems, not to broaden the survey to every form of research automation that touches the pipeline. The relevant question is not whether a system is “more end-to-end” in the abstract, but where its decisive work is located: near literature discovery and drafting, near execution and analysis, near review and rebuttal, or across a broader span of the pipeline. A system's position on this map already says something about the object its architecture must maintain. Systems concentrated around stages 1 and 7 are organized mainly around source selection and evidence-supported passage generation; systems extending through stages 2–7 must maintain an evolving project state that ties together hypotheses, methods, execution traces, and manuscript claims; and systems concentrated around stages 7–8 are organized around an existing draft, reviewer comments, and revision memory.

4.1 Task Coverage as Positional Description

Once the research lifecycle is used in this way, task coverage becomes richer than a simple autonomy scale. Section-level, evidence-grounded writers such as SCHOLARCOPILOT occupy a narrow region linking literature triage to manuscript drafting: their central bottleneck is local claim–citation alignment rather than project-wide scientific closure [72]. Survey and literature-review systems such as SURVEYFORGE and SURVEYX remain largely on the literature-to-manuscript side of the map, but with a broader footprint: the relevant architectural object is now an evolving corpus model and document plan rather than an isolated cited passage [43, 82]. Full-paper systems such as DATA-TO-PAPER and THE AI SCIENTIST shift the center of gravity toward stages 2–7, because the manuscript is expected to package an underlying research process rather than retrieved literature alone [32, 45, 81]. Review, revision, and rebuttal systems such as AGENTREVIEW and SWIF²T, by

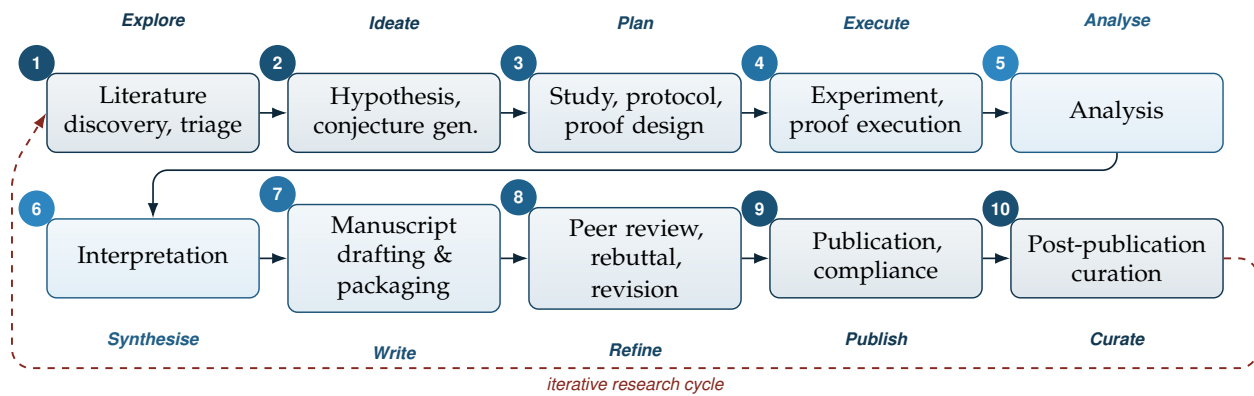


Figure 1. Research lifecycle map contextualizing manuscript-focused systems within the broader research automation pipeline. The dashed return arrow marks the iterative nature of the cycle.

contrast, cluster around stages 7–8, where the operative state is an existing manuscript together with critique and response structure rather than an experimental pipeline [10, 37].

This is why task coverage cannot be reduced to a single scale of autonomy. A system centered on stages 7–8 is not simply a weaker version of one spanning stages 2–7, and a system centered on stages 1 and 7 is not merely an early fragment of an end-to-end agent. These systems are answerable to different scholarly objects, decompose work around different bottlenecks, and fail in different ways. The lifecycle map is useful precisely because it makes those differences visible before one asks how many agents, tools, or loops are involved.

4.2 Artifact Type

Task coverage alone does not determine what kind of scholarly object a system is responsible for. Two systems may occupy nearby regions of the research lifecycle while producing different artifacts and therefore operating under different standards of success and failure. In this survey, four recurrent artifact types organize the literature: section-level evidence-grounded writing, survey and literature-review generation, full-paper manuscript systems, and review, revision, and rebuttal agents. These are the four families surveyed in Section 5. Artifact type matters because it changes what counts as acceptable evidence, what kind of coherence must be maintained, and where correction must occur in the manuscript pipeline.

4.3 Verification Regime

Task coverage and artifact type alone, however, leave a crucial ambiguity. Two systems may occupy similar regions of Figure 1 and produce similar kinds of scholarly artifacts while differing fundamentally in what can verify their principal claims. The third dimension of the taxonomy is therefore the verification regime developed in Section 6: formally verifiable symbolic domains, executable code-and-data domains, tool-mediated physical sciences, and open empirical biological and clinical domains. This axis is not a matter of disciplinary label. It concerns what closes the manuscript’s claims, at what cost and latency, and with what degree of internal versus external control.

Taken together, these three dimensions yield the actual taxonomy. A system spanning stages 3–7 in

an executable code-and-data domain can be organized around reruns, benchmark harnesses, and provenance trails, because much of the verification loop can be closed within the agent pipeline; this is the setting in which architectures such as DATA-TO-PAPER are most intelligible [32]. A superficially similar stage profile in tool-mediated physical science is structured differently, because simulators and instruments act as proxy validators rather than final adjudicators: local success in the tool does not by itself close the manuscript’s claims. In open empirical biological and clinical research, a comparable task footprint must be more conservative still, because decisive validation remains external, slow, and expensive; the architecture therefore shifts toward hypothesis triage, feasibility filtering, and explicit uncertainty rather than strong internal closure. Conversely, in formally verifiable symbolic domains, once formalization succeeds, verifier-coupled search can absorb a much larger share of the loop than is possible in open empirical settings. Similar coverage, then, does not imply similar architecture: the verifier changes what safe autonomy, traceability, and warranted claim strength can mean.

The four families used in Section 5 remain useful because they name recurrent clusters in this space: Family I (section-level evidence-grounded writing), Family II (survey and literature-review generation), Family III (full-paper and end-to-end research agents), and Family IV (review, revision, and rebuttal agents). But these families are descriptive summaries of the current literature, not by themselves a sufficient prospective taxonomy. A newly introduced system is positioned most reliably by specifying its dominant stage footprint, its primary artifact type, and the verifier that governs its core claims.

5 Systems in Practice

This section surveys the most representative research agent systems currently documented in the literature, organized by the four families introduced in Section 4. For each family, we identify the core systems, describe the architectural choices that distinguish them, and characterize the failure modes that follow from those choices. The treatment is intentionally selective: only systems that are analytically informative are discussed in depth.

5.1 Family I: Section-Level Evidence-Grounded Writing

Family I systems operate at the smallest manuscript scale, producing individual scholarly units such as related-work paragraphs, introductions, or method sections under explicit evidence constraints. Unlike higher-level families, they do not attempt to organize or generate a research program as a whole; their aim is to ensure that local claims are supported by appropriate sources.

The defining challenge in this family is therefore not generation itself, but evidence alignment. A system may produce fluent academic prose while attaching weak, irrelevant, or hallucinated citations. The relevant question is not merely whether citations are present, but whether the claims are supported by the right evidence.

SCHOLARCOPILOT [72] is the strongest recent representative of this family. It jointly models text generation and citation retrieval within a unified architecture, thereby reducing the mismatch between what the model writes and the evidence it retrieves. This contrasts with pipeline approaches that treat retrieval and writing as separate stages.

Complementary work such as Şahinuç et al. [57] shows that even in controlled settings, citation-text generation continues to exhibit substantial gaps in attribution precision, underscoring that local grounding remains an unsolved problem.

Benchmarks such as ALCE and CITEBENCH are discussed in Section 7, where they function as evaluation infrastructure rather than as system exemplars.

The characteristic failure mode of this family is citation presence without evidential adequacy: the text appears well supported, while the underlying references do not in fact justify the claims being made.

5.2 Family II: Survey and Literature-Review Generation

Family II systems take a corpus of papers as input and produce a structured long-form synthesis as output. This distinguishes them from Family I systems, which operate at the level of individual sections, and from Family III systems, which aim to produce original research contributions. The central engineering challenge here is not generation per se, but grounded synthesis: covering the relevant literature without sacrificing citation fidelity, handling contradictions across sources, and maintaining structural coherence across a document that may span dozens of sections.

AUTOSURVEY [71] established the modern baseline for this family through a pipeline built around embedding-based retrieval, one-shot outline generation, parallel subsection drafting, and iterative refinement. SURVEYFORGE [82] extends this design with outline heuristics derived from human-written surveys and a memory-driven retrieval agent. SURVEYX [43] separates corpus preparation from generation through an AttributeTree that structures key information before drafting. A more recent line of work identifies the residual weakness of these pipeline systems as local coherence without document-level consistency. SURVEYGEN-I [12] addresses this through adaptive planning and memory-guided writing, while ARISE [73] introduces an iterative, rubric-guided loop with peer-review-based refinement.

Some of the strongest evidence on citation quality comes from systems that frame the task as multi-paper synthesis rather than survey writing alone. OPENSCHOLAR [4] achieves citation accuracy comparable to human experts on ScholarQABench, against a reported GPT-4o hallucination rate of 78–90%. PAPERQA2 [62] has also become one of the most widely used comparative baselines in this space. SURVEYGEN [6] contributes the largest training dataset currently available for this family (4,200+ surveys and 242K references) and shows that fully automatic systems still lag behind humans, especially in citation quality and critical analysis. Secondary systems include LIRA [24], which treats reliability as a first-class design objective; CHATCITE [41], which is relevant for the narrower task of comparative literature summarization; and LLM×MAPREDUCE-V3 [11], which is notable as a modular long-context infrastructure component. Benchmarks for this family are discussed in Section 7.

Family II systems repeatedly perform well on fluency while underperforming on informational value and research guidance. The right diagnosis therefore separates three capabilities that surface metrics often conflate: citation support linkage, conflict resolution across sources, and global structural coherence.

5.3 Family III: Full-Paper and End-to-End Research Agents

Family III systems operate at full-manuscript scale. Many aim to produce complete research artifacts through multi-stage pipelines that combine hypothesis generation, experiment design and execution, analysis, and manuscript writing, while others take completed research materials as input and treat manuscript synthesis itself as the primary task. Unlike section-level or survey systems, their output is a full paper whose claims are expected to reflect an underlying research process rather than a post hoc synthesis. The central challenge in this family is therefore not generation alone, but alignment between narrative and evidence: a system may produce a coherent manuscript even when the underlying experiments are incomplete, biased, or incorrectly interpreted.

The strongest reference systems in this family are THE AI SCIENTIST and THE AI SCIENTIST-V2 [45, 46, 81], DATA-TO-PAPER [32], AI-RESEARCHER [67], and AGENT LABORATORY [58]. A 2026 *Nature* publication provides the first peer-reviewed evidence that an end-to-end system can autonomously generate, execute, and evaluate scientific research across the full pipeline [46]. These systems differ, however, in how tightly manuscript claims are tied to executed evidence. DATA-TO-PAPER enforces backward-traceable links from every numerical value in the manuscript to the specific line of code that produced it, prioritizing auditability over breadth of exploration. THE AI SCIENTIST-V2, by contrast, emphasizes breadth, using agentic tree search to expand hypothesis-space coverage; it also produced the first fully AI-generated manuscript to pass peer review at an ICLR 2025 workshop. AI-RESEARCHER introduces Resource Analyst agents that explicitly map mathematical formulations to code implementations before experimentation, thereby reducing hallucination risk. AGENT LABORATORY instead delegates initial ideation to the human researcher while automating the downstream pipeline.

A complementary recent direction studies full-manuscript generation from completed research materials rather than from a closed experimental loop. PAPERORCHESTRA transforms unconstrained pre-writing inputs, including idea summaries, experimental logs, and venue-specific templates, into submission-ready LaTeX manuscripts with literature synthesis and generated visuals, using specialized agents for outline construction, literature review, plot generation, section writing, and iterative refinement [63]. Analytically, it belongs in Family III because its primary artifact is a full paper rather than a section or survey, but it should be distinguished from end-to-end research agents: the underlying research process is assumed through provided materials rather than autonomously executed and validated within the system itself. Its companion benchmark, PAPERWRITINGBENCH, is discussed in Section 7.

Recent systems suggest that this capability is not confined to a single lineage. ZOCHI [35] reports acceptance at an ACL 2025 main-track venue with minimal human intervention, while systems such as EVOSCIENTIST and KOSMOS explore longer-horizon coordination and more persistent research processes. Evidence across these systems remains uneven, however, and not all provide comparable support for end-to-end manuscript quality.

A broader set of systems contributes to the surrounding landscape with varying degrees of completeness, including DEEPSCIENTIST, AUTORESEARCHCLAW, DOLPHIN, ALETHEIA, TINYSOCIENTIST, DENARIO, and INTERNAGENT [25, 34, 69, 77, 79, 83, 84]. These systems are informative for coverage, but they do not yet provide evidence comparable in strength to the core systems above. Several systems lie at the boundary of this family. AI CO-SCIENTIST [26] focuses primarily on

hypothesis and proposal generation rather than full manuscript production. FARS [2, 3] presents an end-to-end pipeline through public documentation, but without reproducible technical detail. AUTORESEARCH and AUTORESEARCH@HOME [39, 48] are better understood as autonomous experimentation loops whose primary artifacts are code modifications and performance changes rather than manuscripts.

A related class of systems operates directly on an existing manuscript state rather than generating one from scratch, including PAPERDEBUGGER, BIBBY, DORA, and CITELLM [29, 30, 33, 36]. Their strength lies in state-aware correction under project constraints; their limitation is that local improvements do not guarantee global scientific validity. For that reason, we treat them as a subcase within this family rather than as a separate family of systems.

The characteristic failure mode of Family III is that narrative confidence can outrun verification depth: a system may produce a convincing research paper whose claims are not fully supported either by the underlying research process or by the materials from which the manuscript is assembled. Detecting this failure therefore requires inspection beyond the manuscript alone: independent re-execution for execution-coupled systems, or audit of the supplied research materials together with external validation where necessary for systems that write from completed inputs.

Recent systems also point toward emerging directions in this space. PAI already operates on manuscript-like artifacts, while platforms such as PSI and SKYDISCOVER, together with recent proposals for agent-native scientific ecosystems [1, 7, 54, 75], extend the paradigm toward persistent, multi-agent research environments.

5.4 Family IV: Review, Revision, and Rebuttal Agents

Family IV systems make critique an explicit component of the generation process. Rather than producing content directly, they evaluate, refine, or contest existing drafts through structured feedback. Their primary role is therefore a form of approximate verification through critique: identifying weaknesses, suggesting revisions, or simulating peer-review dynamics. The central challenge is that critique is not equivalent to verification. Systems can produce detailed and plausible feedback without increasing the probability of detecting real errors, leading to the characteristic risk of self-confirming criticism, in which iterative review improves fluency and presentation while leaving substantive mistakes unchanged.

Two main design patterns recur in this family. The first simulates the peer-review process itself. AGENTREVIEW [37] models reviewers, authors, and area chairs as interacting agents, and finds that 37.1% of acceptance-decision variation is attributable to reviewer bias rather than paper quality. REVIEWAGENTS [22] extends this paradigm through structured training on large-scale review data and a multi-role architecture that includes reviewer and area-chair agents.

The second pattern treats critique as a tool for improving manuscripts or existing reviews. Systems such as MARG [14], DEEPREVIEW [86], and CYCLERESEARCHER/CYCLEREVIEWER [76] explore multi-agent generation and reconciliation of reviewer-style comments. SWIF²T [10] focuses instead on fine-grained, paragraph-level feedback targeting specificity and actionability, while AUTOREBUTTALCLAW [44] automates the rebuttal-writing phase specifically by profiling individual reviewers and generating venue-compliant responses for major machine learning conferences.

The strongest empirical evidence for real-world impact comes from two deployment studies. Liang et al. [42] evaluated GPT-4 feedback on more than 4,800 papers and found that 82.4% of surveyed researchers rated it as more beneficial than at least some human reviews. At ICLR 2025, a Review Feedback Agent was deployed on 20,000 reviews [68]; 27% of reviewers updated their reviews, incorporating more than 12,000 suggestions, with measurable improvements in review length and actionability. The foundational dataset benchmark for this family remains Yuan et al. [85]’s ASAP-Review, with sentence-level aspect annotations derived from ICLR and NeurIPS reviews.

The defining failure mode of Family IV is the conflation of preference with correctness. Systems may produce critiques that are fluent, detailed, and aligned with human expectations, yet still fail to identify substantive scientific errors. Effective evaluation must therefore distinguish critique that improves perceived quality from critique that increases epistemic reliability.

6 Domain-Specific Applications

The families introduced in Section 5 organize research agent systems by the kind of artifact they produce and by the portion of the research pipeline they cover. This organization is intentionally agnostic to scientific domain: a system belongs to a given family regardless of whether it operates in machine learning, biology, or formal mathematics. That perspective is useful for identifying recurring architectural patterns, but it leaves open a complementary question.

Scientific domains differ in how claims can be supported, tested, and falsified. These differences are not superficial. They determine what counts as evidence, how feedback can be obtained, and which forms of validation are practically available. As a result, an architecture that is well-posed in one domain may become unreliable, inefficient, or even incoherent in another.

This section examines those constraints and their architectural implications. The relevant unit of analysis is not the disciplinary label attached to a manuscript, but the *verification regime* governing its core claims: how those claims can be checked, at what cost and latency, and whether verification can be closed within the agent pipeline or instead depends on external processes outside the system’s control. We consider four such regimes: formally verifiable symbolic domains; executable code-and-data domains; tool-mediated physical sciences, in which instruments or simulators act as proxy validators; and open empirical biological and clinical domains, in which decisive validation remains external to the agent loop. These regimes are selected not because they exhaust the space of scientific practice, but because they impose structurally distinct pressures on system design.

Where useful, we also draw on systems whose primary artifact is not a manuscript but an experimental design, a simulation output, or a laboratory protocol. Such systems are not treated as core comparison objects in the survey; rather, they serve as evidence for the architectural pressures characteristic of a given verification regime.

The aim of this section is therefore not to resurvey systems within each regime, since those are discussed in Section 5 where relevant. It is to make explicit the design pressures that arise when manuscript-producing agents are instantiated in scientific settings with fundamentally different verification structures.

These regimes cut across conventional disciplinary boundaries, and a single manuscript may span more than one of them. For analytical purposes, however, the relevant assignment is determined

by the bottleneck verifier of the manuscript’s principal claims: the mechanism that ultimately determines whether those claims can be accepted, rejected, or only weakly supported. Computational biology, for example, often falls into the executable code-and-data regime when its central claims are grounded in reproducible workflows over existing datasets, whereas wet-lab and clinical biomedicine fall into the open empirical regime when decisive validation depends on new experiments or prospective studies. The relevant comparison is therefore between verifier types rather than between department names.

6.1 Formally Verifiable Symbolic Domains

In formally verifiable symbolic domains, the decisive constraint is that a claim becomes admissible only once it has been translated into a formal language accepted by a proof assistant. This regime includes classical formal mathematics, but it also extends to parts of theoretical science whenever a manuscript’s core contribution can be compiled into machine-checkable symbolic statements. The central architectural fact is therefore that verification is not an external evaluation step appended after generation; it is the environment within which meaningful progress occurs. Systems such as ALPHAPROOF [19] and APOLLO [51] make this explicit by coupling language-model reasoning directly to Lean and related theorem-proving environments.

The epistemic structure of this regime is deductive. A valid claim is one that follows from explicit assumptions within a formal system, and verification is exact once the claim has been successfully formalized. The evidence substrate is correspondingly symbolic: theorem statements, proof states, retrieved lemmas, compiler errors, tactic traces, and cross-file dependencies. The main bottleneck is therefore not noisy evidence or expensive experimentation, but the translation of informal mathematical content into formal objects with the correct definitions, library dependencies, and tactic structure. Without that translation step, which remains difficult even for strong systems, the verification mechanism cannot be engaged at all.

Architecturally, this shifts the balance strongly toward execution-backed search. Planning still matters, but primarily in the form of proof decomposition: selecting auxiliary lemmas, retrieving relevant library facts, choosing proof strategies, and deciding when to repair or backtrack. Once that structure has been proposed, progress is governed by the verifier rather than by free-form self-critique. APOLLO [51] is especially informative on this point: it uses compiler-guided repair to decompose failing proofs into sublemmas and iteratively reclose them, precisely because verifier feedback is treated as a first-class signal rather than as an optional check. In practical terms, proof-assistant coupling, retrieval over formal libraries, and verifier-guided repair are necessary; human judgment about novelty or mathematical significance remains optional; and prose-only self-critique without a machine verifier is structurally unreliable. Autonomous verification loops can therefore often be closed within the agent pipeline once a claim has been formalized, although humans remain useful for selecting conjectures, introducing new definitions or axioms, and deciding whether a formally verified result is interesting enough to merit inclusion in a manuscript.

The dominant failure mode of a domain-agnostic manuscript agent in this regime is the generation of linguistically plausible but formally invalid proof sketches. A model optimized for natural-language fluency can produce text that reads like a coherent mathematical argument while silently omitting, misstating, or hallucinating proof steps that do not compile. The failure is largely invisible to surface-level review and becomes visible only once formal verification is brought into the loop,

which is precisely the mechanism a domain-agnostic architecture lacks.

6.2 Executable Code-and-Data Domains

In executable code-and-data domains, the dominant constraint is that substantive claims should reduce to an executable analytical trace: code, data, configuration, and an observable outcome. This regime is exemplified most clearly by AI engineering and computational research, but it also includes the computational side of the life sciences whenever a manuscript’s core claims are grounded in analysis over existing datasets rather than in new physical interventions. Systems such as THE AI SCIENTIST [45, 81], AGENT LABORATORY [58], and DATA-TO-PAPER [32] operate in this regime at the manuscript level, while CELLAGENT [80] and BIODSA-1K [74] show that the same structural pattern extends into computational biology.

The epistemic structure here is empirical and statistical rather than deductive. A valid claim is typically comparative, predictive, or descriptive: that a method improves a benchmark, that a pattern appears in a dataset, or that an analysis supports or fails to support a hypothesis. Verification is largely internal to the agent pipeline, because the agent can rerun code, regenerate figures, recompute metrics, and compare outputs against baselines. The evidence substrate is therefore executable: repositories, scripts, datasets, random seeds, tables, logs, and benchmark harnesses. This makes provenance unusually important. DATA-TO-PAPER [32] is built explicitly around backward traceability from manuscript claims to the code lines that generated them, and that design choice is structurally important in a regime where generation can otherwise outrun auditability.

Architecturally, this regime favors a relatively balanced planner–executor design. The system must decompose the research problem, but it must also repeatedly instantiate those plans in a sandboxed computational environment. Code execution, benchmark harnesses, analysis tools, and provenance tracking are therefore not optional embellishments but core architectural components whenever the goal is to verify rather than merely narrate results. In practical terms, executable analysis, rerunnable evaluators, and artifact-level provenance are necessary; human feedback on framing, novelty, and error interpretation remains optional but often valuable; and one-shot manuscript generation without reruns or audit trails is structurally unreliable. Autonomous verification loops are feasible here, and this distinguishes the regime from the open empirical one, but the evidence does not support the stronger claim that human oversight becomes irrelevant. AGENT LABORATORY [58] finds that human feedback at each stage materially improves research quality, and DATA-TO-PAPER [32] likewise treats human co-piloting as increasingly valuable as goals and datasets grow in complexity.

The dominant failure mode in this regime is a capability–reliability gap coupled to metric and narrative inflation. An agent can produce code that runs and figures that look convincing while the underlying hypothesis generation and experimental design remain shallow. Recent full-pipeline evaluations are consistent on this point. PAPERBENCH [64] shows that faithful replication of recent machine-learning papers remains substantially harder for current agents than for human experts; SCIENCEAGENTBENCH [13] reports low independent solve rates on data-driven discovery tasks across disciplines; and OpenAI’s FRONTIERSCIENCE [50] similarly finds a large gap between structured scientific problem-solving and open-ended research reasoning. BIODSA-1K [74] adds an especially important complication in the biomedical portion of this regime: some hypotheses

are non-verifiable from the available data, so even a well-executed computational pipeline must sometimes conclude not that a claim is false, but that the evidence substrate is insufficient to adjudicate it.

6.3 Tool-Mediated Physical Sciences

In tool-mediated physical sciences, the core constraint is that claims are evaluated through simulators, workflow engines, or physical instruments whose outputs are informative but remain only proxy validators of the final scientific conclusion. What unifies this regime is not a shared disciplinary label, but a shared mediation structure: the agent must compile scientific intent into simulator inputs, workflow specifications, or instrument commands, and then interpret telemetry, images, spectra, or measurement outputs as evidence for a claim that ultimately concerns the physical world.

Unlike the other three regimes, the systems that most clearly expose the architectural pressures of this setting do not generally produce scholarly manuscripts as their primary artifact. GRACE [28], COSCIENTIST [8], A-LAB [66], and AILA [47] are instrument- or simulation-native systems whose primary outputs are experimental designs, synthesized compounds, or laboratory results rather than papers. They matter here not as core comparison objects for the survey, but as evidence for the verification pressures characteristic of this regime. The current scarcity of manuscript-first systems in this space suggests, although it does not by itself prove, that bridging from a locally closed experimental loop to a manuscript grounded in that loop is harder here than in purely computational research.

The epistemic structure of this regime is hybrid. Claims are often mechanistic, numerical, and empirical at once: a detector design improves a simulated objective under physical constraints; a synthesis route produces a compound with target properties; a laboratory control sequence yields a reproducible physical state. Verification can often be closed locally, in the sense that the agent can rerun the simulator or instrument and inspect the result, but local closure should not be mistaken for final truth. A simulator objective, an X-ray diffraction match, or a microscopy output is much stronger evidence than a text-only conjecture, yet each remains only a partial validator of a broader scientific claim.

Architecturally, this favors nested loops rather than a single flat planner–executor structure. An inner loop executes fast local optimization or control over the simulator or laboratory platform, while an outer loop must interpret whether local success in the tool actually corresponds to the intended scientific question. SAGA [15] makes this pressure explicit by treating fixed quantitative objectives as potentially imperfect scientific proxies and arguing that objective critique must itself become part of the architecture. External tools are therefore mandatory not merely as passive resources but as the substrate on which claims are instantiated. In practical terms, tool adapters, workflow compilers, and proxy-aware interpretation layers are necessary; fully manuscript-first narration is optional and currently secondary; and fixed-objective optimization detached from proxy critique is structurally unreliable. The literature already contains convincing examples of substantial local autonomy: GRACE [28] can execute multi-step simulation workflows with full provenance tracking, and A-LAB [66] closes the synthesis-to-characterization loop for inorganic materials without human intervention during execution. Human checkpoints nonetheless remain advisable at objective selection, safety gating, calibration, and the release of final claims, because

those decisions operate at a level above local tool success.

The characteristic failure mode of a domain-agnostic manuscript agent in this regime is proxy collapse: mistaking success in the simulator or apparatus for scientific success. AILA [47] is especially informative here because it documents the “sleepwalking” phenomenon directly: agents operating an atomic force microscope deviated from stated instructions and carried out unauthorized steps after prompt variation, and the authors explicitly show that strong performance on materials-science question answering does not transfer to reliable laboratory execution. The gap between domain knowledge and operational competence under tool coupling is therefore not merely a performance problem, but a structural one.

6.4 Open Empirical Biological and Clinical Domains

In open empirical biological and clinical domains, the decisive constraint is that the final verifier lies outside the agent. The system may generate hypotheses, experimental designs, literature syntheses, or manuscript drafts, but decisive validation still depends on wet-lab experiments, organoids, animal models, patient cohorts, or clinical procedures that are costly, slow, ethically regulated, and only partially automatable. This is the regime in which the difference between computational support and actual scientific closure is greatest. AI CO-SCIENTIST [26] makes this explicit by framing its output as biomedical hypotheses and proposals that subsequently undergo experimental validation, and BIODISCOVERYAGENT [56] likewise focuses on designing new perturbation experiments rather than claiming that the loop is already closed within the agent itself.

The epistemic structure here is hybrid and often causal. A valid claim may concern a mechanism, an intervention, or a biological target, but the evidentiary route to that claim is distributed across literature, prior datasets, assay outputs, and new experiments whose outcomes are not available at generation time. Verification is therefore external, high-latency, and expensive relative to the previous regimes. The evidence substrate is correspondingly heterogeneous: papers, cohort data, omics measurements, protocols, assay outputs, interventions, and, in clinical settings, regulatory and ethical constraints on what can be tested and how quickly. Even when automation can accelerate parts of the pipeline, the final verifier typically remains outside the writing loop itself.

Architecturally, this shifts the balance strongly toward planning, triage, and uncertainty management. Because physical execution is scarce and expensive, the system should invest more effort in filtering hypotheses, checking novelty and feasibility, identifying what is and is not testable from existing evidence, and producing protocols or ranked experimental programs rather than pretending to complete validation internally. Autonomous subloops remain possible over existing data or bounded validation procedures. POPPER [31] shows that falsification-oriented validation over measurable implications can be automated with statistical error control, and BIODSA-1K [74] shows that biomedical hypothesis validation can be benchmarked over realistic data-analysis tasks. But these remain partial closures. In practical terms, hypothesis triage, feasibility filtering, and explicit representation of uncertainty are necessary; autonomous subloops over existing data are optional but valuable; and any architecture that treats literature synthesis plus existing data analysis as equivalent to decisive biological validation is structurally unreliable. Human checkpoints therefore remain structurally necessary at protocol approval, intervention selection, and the release of strong causal claims whose support depends on external experimental or clinical validation.

The dominant failure mode in this regime is speculative manuscript inflation: a domain-agnostic manuscript agent can assemble a coherent biological narrative from literature and partial data while overstating what has actually been validated. BIODSA-1K [74] is particularly informative because it includes non-verifiable hypotheses, forcing the system to distinguish between false claims and claims that cannot be adjudicated from the available data. In open empirical biology and medicine, that distinction is not a marginal detail, but part of the architecture itself: a system that cannot represent non-verifiability will systematically overclaim.

Cross-regime synthesis. Across all four regimes, the manuscript shell is more stable than the verifier core. Retrieval, decomposition, persistent state, provenance tracking, drafting, and internal critique recur widely, from DATA-TO-PAPER [32] and AGENT LABORATORY [58] to THE AI SCIENTIST [45, 81] and AI CO-SCIENTIST [26]. What changes structurally from regime to regime is the object that closes the loop on the manuscript’s core claims: a proof assistant in formally verifiable domains; an executable benchmark or analysis harness in code-and-data domains; a simulator or instrument stack, whose outputs function only as proxy validators, in tool-mediated physical sciences; and an external experimental program in open empirical biology and medicine. The nearer the verifier is to exact and internal, the more autonomy can safely migrate into the agent loop. The farther it is from the agent, and the higher its cost and latency, the more the architecture must front-load planning and preserve explicit human checkpoints. The central failure of domain-agnostic manuscript agents is therefore not simply hallucination in the ordinary language-model sense, but verifier mismatch: the system makes claims whose validation path is weaker, slower, or more indirect than its architecture assumes.

7 Benchmarking and Evaluation

Evaluation of manuscript-producing research agents has not yet converged on a single benchmark ecosystem with a shared scale of success. Existing benchmarks occupy only partially equivalent regimes, and scores across them are therefore not directly comparable. At a minimum, section- and survey-level writing quality, full-paper pipeline quality, and review or revision utility must be separated analytically. A strong result on ALCE or CITEBENCH is evidence about local citation-grounded generation; a strong result on PAPERBENCH, SCIENCEAGENTBENCH, or FRONTIER-SCIENCE is evidence about execution-backed research competence; a strong result on MMREVIEW is evidence about review generation or critique utility. These are not different points on a single ladder of manuscript quality. They evaluate different artifacts under different verification assumptions, and they reward different forms of success.

This distinction matters because the object that ultimately matters in this survey is not language quality in the abstract, but the credibility of a manuscript as a scientific artifact. A benchmark can score fluency, citation attachment, code execution, or review plausibility without ever asking whether a manuscript’s central claims are proportionate to its evidence. The current landscape is therefore better understood as a fragmented set of proxy regimes than as a unified framework for evaluating scientific writing.

7.1 Section-Level Grounding and Survey Synthesis

In the first regime, benchmarks evaluate whether generated scholarly text is grounded in sources and whether longer syntheses are genuinely informative. ALCE [21] and CITEBENCH [20] are most useful as tests of citation-conditioned generation and local support. HALOGEN [55] and the SCIHAL25 shared task [60] move further toward scientific factuality by checking atomic claims and claim-level support in research-assistant outputs. SURVEYBENCH [65] extends this regime from paragraph-scale grounding to long-form synthesis by asking whether a generated survey is structured, answerable, and information-rich rather than merely fluent. What this regime measures, then, is primarily local evidence linkage, citation presence, topical coverage, and certain aspects of synthesis depth.

What it does not measure is equally important. These benchmarks do not establish whether the cited literature is the right literature rather than merely some literature, whether the strength of a claim is warranted by the evidence invoked, or whether a survey accurately represents disagreement within a field. A generated review may therefore be densely cited, topically broad, and highly readable while still being epistemically distorted. That distortion is not a superficial flaw. If an agent treats a weak source as decisive evidence, omits limiting studies, or smooths over conflict across papers, the resulting text may satisfy the benchmark while still misrepresenting the state of knowledge. In this regime, current evaluation is better at testing citation attachment than citation reasoning.

The distinction between section-level and survey-level evaluation should also not be reduced to a simple matter of length. Section benchmarks ask whether a local claim can be supported sentence by sentence. Survey benchmarks ask whether a document can teach, organize, and synthesize a field. These are related but non-identical capacities. A system may have excellent local grounding and still fail at long-range synthesis, or it may produce a plausible survey narrative whose individual support links are weak. The move from section quality to survey quality therefore changes the artifact being evaluated, not merely the scale of the output.

7.2 Full-Paper Pipeline Quality

The second regime has so far been closer to evaluating research execution than to evaluating manuscripts as integrated documents. PAPERBENCH [64], SCIENCEAGENTBENCH [13], and FRONTIERSCIENCE [50] are especially important because they expose the limits of current systems under external checking. They test whether an agent can replicate work, generate executable analyses, or reason through hard scientific tasks under constrained rubrics. For manuscript-producing agents operating in executable domains, these are much stronger signals than fluency metrics, precisely because they force the system into contact with an external verifier.

At the same time, these benchmarks are not direct evaluations of full-paper quality. The evaluated object is usually a codebase, execution trace, or short-form answer rather than the manuscript itself. A benchmark can tell us whether an experiment ran, whether a method was reproduced, or whether a reasoning trajectory satisfied a rubric. It usually cannot tell us whether the abstract accurately states the contribution, whether the discussion overgeneralizes beyond the evidence, whether caveats are propagated consistently across sections, or whether the paper's argumentative structure is honest about uncertainty. In that sense, this regime measures upstream research competence

while largely treating the manuscript as an unevaluated byproduct.

A notable recent counterpoint is PAPERWRITINGBENCH, introduced together with PAPERORCHESTRA, which evaluates full-manuscript generation from reverse-engineered pre-writing materials rather than from an autonomous research loop [63]. The benchmark contains 200 top-tier AI conference papers (100 each from CVPR 2025 and ICLR 2025) and provides an idea summary, an experimental log, and venue-specific L^AT_EX templates and guidelines, thereby isolating drafting from completed research. This moves evaluation closer to the manuscript as an object, and the accompanying side-by-side human study reports that PAPERORCHESTRA outperforms autonomous baselines by 50%–68% in literature-review quality and 14%–38% in overall manuscript quality. But the benchmark still assumes that the underlying scientific process has already occurred; it does not test whether the agent autonomously generated, executed, or validated the research program whose claims it writes up.

That remaining gap matters for scientific credibility rather than presentation alone. A manuscript is not a stylistic wrapper around execution. The inferential move from result to claim is itself part of the scientific contribution and part of the failure surface. A system may run analyses correctly and still write a paper that overstates what was validated, suppresses non-verifiability, or presents exploratory results as settled findings. Conversely, a manuscript may appear coherent while the underlying pipeline is shallow or irreproducible. Full-paper evaluation therefore remains underdeveloped not because manuscript-focused benchmarks are absent, but because current benchmarks still assess writing quality and upstream research validity more easily in isolation than as a single integrated scientific artifact.

7.3 Review and Revision Utility

The third regime evaluates manuscript-producing agents as critics, editors, or revisers. Here the question is not whether the agent can produce a paper from scratch, but whether it can improve one. MMREVIEW [23] is the clearest reference point among the benchmarks used in this survey, and more broadly this regime tends to score review quality through human agreement, aspect coverage, actionability, or related proxies. These are meaningful signals: they capture whether a review appears plausible, whether it touches the kinds of issues human reviewers tend to notice, and whether authors are likely to experience it as useful.

But review and revision utility are not the same as scientific error correction. A fluent review that misses a fatal methodological flaw, attacks an incorrect premise, or improves wording without improving correctness can still look strong in this regime. The same is true for revision systems that patch references or paragraphs locally: they may increase local validity while leaving the global thesis unchanged, or even less coherent than before. What current evaluation often captures is the usefulness or plausibility of critique, not demonstrated improvement in the reliability of the manuscript after critique.

This matters especially because review is often treated as a safeguard for the scientific record. If the safeguard is itself evaluated mainly by stylistic agreement with human reviews, the benchmark risks measuring whether the system can imitate peer-review discourse rather than whether it can detect consequential error. For manuscript-producing agents, that is a serious distinction. A review benchmark that rewards plausible criticism without establishing factual correctness is still

evaluating discourse quality more directly than scientific trustworthiness.

7.4 The Unresolved Object of Evaluation

Across all three regimes, three missing dimensions recur. First, no current benchmark measures global manuscript coherence. For the systems studied here, this means more than generic discourse smoothness. The unresolved question is whether the abstract, introduction, methods, results, discussion, and limitations sections remain scientifically consistent with one another, whether claims are tightened or weakened appropriately as evidence changes, and whether the document as a whole sustains a single warranted argument. A manuscript can be locally well written and still globally unsound.

Second, no current benchmark measures contradiction handling across sources. This is most obvious for survey and literature-review agents, but it also matters for related-work sections and revision systems. Scientific credibility often depends on whether an agent notices disagreement, distinguishes robust findings from contested ones, and explains why sources diverge. A benchmark that rewards coverage without adjudication can therefore reward a polished erasure of genuine scientific conflict.

Third, no current benchmark measures the match between validation mechanism and claim substrate. This is a cross-regime problem rather than a local metric failure. Claims grounded in proofs, executable analyses, instruments, or external experiments cannot be judged by the same validator without distortion. A citation-entailment score is too weak for an empirical performance claim; code execution is too weak for a biological mechanism that ultimately requires external experimental validation; human review agreement is too weak for establishing factual correctness. When the validator is weaker than the claim requires, the system can appear better evaluated than it truly is.

The deeper open question is therefore not only how to build better benchmarks, but what the proper object of evaluation should be in principle. It may be the final manuscript, but it may also need to be a richer artifact: manuscript plus provenance trace, claim–evidence graph, execution record, or revision history. It may require different evaluators for different claim types within the same paper rather than a single scalar score. It may also require accepting that some dimensions of manuscript quality are only partially automatable because they concern scientific judgment rather than textual regularity. What already seems clear is that benchmark results will remain easy to overinterpret until evaluation is organized around the manuscript as an epistemic artifact rather than around whichever proxy is cheapest to score.

8 Structural Limits, Epistemic Ownership, and Institutional Choice

The evaluation gaps identified in Section 7 are symptoms of a deeper unresolved structure. Manuscript-producing agents are asked to generate artifacts whose scientific validity is global, while most of the mechanisms currently available to constrain them operate locally. A manuscript is not a sequence of independently acceptable spans, but a structured claim about what was asked, what was done, what was found, and what follows from it. The central open question is therefore not only how these systems should be scored, but whether present architectures can compose local grounding, execution, and critique into global manuscript validity, or whether they are approaching

the limit of what can be achieved by chaining together locally validated operations.

Current systems are strongest when the task can be decomposed into units with accessible feedback: retrieve the relevant paper, execute the benchmark, repair the paragraph, answer the reviewer. What remains unclear is whether the same decomposition can support the validity of the document as a whole. Global validity depends on relations among sections: whether the abstract says only what the results support, whether limitations discovered late in the pipeline propagate back to the contribution claim, whether negative or ambiguous evidence survives into the discussion rather than being smoothed away. These are not simply longer-context versions of sentence-level fidelity. They suggest that manuscript reliability may be a representational problem as much as an optimization problem. A system that stores a project as context windows, memories, and drafts may never fully represent the dependency structure that makes a paper scientifically sound. If that diagnosis is correct, then a materially different generation of systems will require explicit claim states, evidential typing, and revision-aware document models rather than simply larger context windows or more agents.

A related unresolved question concerns the relation between autonomy and closure. Throughout this survey, safe autonomy has expanded most readily where strong verification is available inside the agent loop. Where the decisive verifier is external, slow, or only partially automatable, the architecture can become rhetorically complete before it becomes scientifically warranted. It remains open whether this is primarily an engineering lag or a structural ceiling. It may be that full manuscript autonomy is attainable only in regimes where core claims can be checked internally, whereas in open empirical domains the honest role of the system is narrower: prioritizing hypotheses, organizing evidence, and drafting claims whose final status remains undecided. On that reading, the hard problem is not making agents more assertive, but making them capable of writing without falsely implying closure.

This distinction also reframes how failure should be theorized. Many manuscript errors are described as hallucinations, but at the document level the deeper problem is often closure failure: the system resolves scientific uncertainty linguistically before science resolves it materially. Better models may reduce fabricated citations or local factual errors, yet still intensify this broader failure if fluency, compression, and narrative smoothness improve faster than verification. A system that writes more convincingly is not necessarily a system that knows more; it may simply be better at turning partial evidence into finished prose. The next generation will be qualitatively different only if it can preserve the distinction between what is established, what is merely compatible with current evidence, and what still depends on external adjudication.

These technical limits lead directly to a second unresolved problem: what becomes of authorship and intellectual ownership when manuscripts are generated rather than written? Traditional scientific norms assume a legible path from reading, analysis, interpretation, and drafting to a responsible human author. That path can be collaborative and socially distributed, but it remains, at least in principle, reconstructable. Manuscript-producing agents disrupt this assumption because they do not merely polish wording. They can supply structure, introduce framings, surface citations, and fill inferential gaps in ways that feel like seamless assistance while still altering the intellectual content of the paper. The resulting manuscript is therefore not merely co-produced; it is co-produced under conditions in which the boundaries of contribution are unusually hard to recover after the fact.

This is why provenance is more than a disclosure problem. One part of the difficulty arises when generated text draws on retrieved or tool-accessed sources that shaped the manuscript substantively but are not reflected in the final citation record. In such cases, the intellectual lineage may in principle be reconstructable from pipeline logs even when it is invisible in the paper itself. A second difficulty is harder: some framings, comparisons, or inferential moves may enter through model priors or opaque generation dynamics rather than through identifiable source access. Here the manuscript may benefit from prior intellectual labor whose lineage is not recoverable from the final text, and perhaps not recoverable from the generation pipeline in any robust sense. The literature has begun to describe this broader problem in terms of provenance or hidden intellectual debt [16]. Existing authorship conventions can record that a tool was used and can insist that humans remain answerable for the paper, but they do not specify which parts of the argument originated in human judgment, which were retrieved from identifiable sources, and which entered through opaque model priors or generation heuristics. For manuscript-producing systems, attribution and accountability are therefore linked: if the path by which an idea entered the manuscript cannot be reconstructed, responsibility for that idea also becomes harder to locate in anything more than a formal sense.

The mismatch extends to reproducibility. In ordinary scientific practice, reproducibility attaches to data, code, protocols, and analysis pipelines. For manuscript agents, however, an important part of the epistemic work also occurs in retrieval states, prompt histories, memory stores, tool traces, model versions, and revision loops that are rarely preserved in a contestable form. A paper may therefore be experimentally reproducible while remaining procedurally opaque as a generated artifact. The inverse is also possible: a generation can be replayed without establishing that the claims it expresses were warranted. The open question is not whether every token of model output should be logged, but what level of provenance is sufficient for generated scientific writing. Disclosure without reconstructable claim lineage is too weak if manuscript agents make substantive epistemic contributions, yet full pipeline transparency may be impractical. Systems will become meaningfully different only if they can expose enough of the path from source to claim to make later contestation and credit assignment possible.

Once these systems are understood in this way, the institutional issues are not secondary add-ons but direct consequences of the technical analysis. Journals, conferences, funding agencies, and research communities will have to decide whether they are evaluating a text, a process, or both. They will have to make these decisions before there is stable evidence about which disclosure or audit regimes actually preserve reliability, which means that early local policies may harden into field-wide norms faster than the evidence base matures. If manuscript agents remain stylistic assistants, conventional disclosure may be enough. If they increasingly shape literature review, experimental interpretation, reviewer response, and the final scaling of claims, then evaluation of the finished prose alone becomes insufficient. Institutions will need to determine whether some uses are acceptable only when accompanied by auditable provenance, whether externally unverified claims should be marked differently when drafted through agentic pipelines, and whether reviewer-side use can coexist with confidentiality and independent judgment. A community in which authors optimize for agent-legible review criteria and reviewers rely on similar systems to generate critique risks converging on a closed loop of mutually intelligible but jointly ungrounded evaluation.

For that reason, treating systems as non-authors and retaining human responsibility should be

understood as a baseline rather than a complete solution. What remains unsettled is which disclosure, audit, or deployment standards could track these system properties closely enough to preserve meaningful accountability. If manuscript-producing agents materially shape literature review, interpretation, and claim formation, then responsibility that cannot be exercised through inspection, contestation, and selective verification risks becoming nominal rather than real. The broader implication is that manuscript-producing agents force science to confront the manuscript not as a neutral report of research, but as part of the research mechanism itself. Until systems are designed around that fact, further progress will likely produce faster and more fluent research agents, but not necessarily agents whose manuscripts deserve stronger scientific trust.

9 Conclusion

This survey argues that manuscript-producing research agents are not best understood as a single technology advancing uniformly in agent count or along one autonomy scale. The more informative comparison is lifecycle coverage plus verification regime. Current systems are strongest when they are designed around definite manuscript obligations and when their claims are tied to the strongest verifier available to them. Retrieval, planning, execution, critique, memory, and multi-agent coordination all matter, but only when they are attached to real evidential bottlenecks.

This perspective also changes how progress in the field should be read. Section writers, survey generators, full-paper pipelines, and review agents are not points on a single maturity ladder. Likewise, current benchmark ecosystems are not a unified measure of manuscript quality. Scores on citation grounding, execution-backed research competence, and review utility remain fragmented proxies that test different artifacts under different validators. The field is therefore uneven rather than uniformly immature or uniformly advanced: autonomy expands furthest where verification is internal, exact, or cheaply rerunnable, and narrows sharply where the decisive verifier is slow, costly, indirect, or external.

What remains unresolved is more structural than incremental. Current systems can often secure local grounding, local execution, or local critique without guaranteeing global manuscript validity. Whether that gap can be closed will likely require richer document representations in which claim status, uncertainty, evidential support, and revision dependencies are explicit rather than implicit. It is likewise unresolved whether open empirical domains admit full manuscript autonomy at all, or whether honest automation there is structurally narrower than current rhetoric suggests. Provenance, authorship, and institutional governance are downstream consequences of this same technical gap.

The clearest lesson is that stronger scientific trust will not come from making these systems merely larger, more fluent, or more agentic. It will come from architectures that represent claim status explicitly, propagate uncertainty across the manuscript as a whole, expose a contestable path from source, execution, or experiment to final prose, and avoid *closure failure*, that is, the conversion of partial verification into the appearance of settled science.

References

- [1] Mahmoud Abdelmoneum, Pierfrancesco Beneventano, and Tomaso Poggio. pai/msc: ML theory research with humans on the loop. <https://poggioai.github.io/papers/poggioai-msc-v0.pdf>, 2026. Accessed: 2026-03-25.
- [2] Analemma. Introducing FARS. <https://analemma.ai/blog/introducing-fars/>, 2026. Official blog post, accessed March 19, 2026.
- [3] Analemma. FARS. <https://analemma.ai/fars>, 2026. Official product page, accessed March 19, 2026.
- [4] Akari Asai, Jacqueline He, Rulin Shao, Weijia Shi, Amanpreet Singh, Joseph Chee Chang, Kyle Lo, Luca Soldaini, Sergey Feldman, Mike D’arcy, David Wadden, Matt Latzke, Minyang Tian, Pan Ji, Shengyan Liu, Hao Tong, Bohao Wu, Yanyu Xiong, Luke Zettlemoyer, Graham Neubig, Dan Weld, Doug Downey, Wen tau Yih, Pang Wei Koh, and Hannaneh Hajishirzi. Openscholar: Synthesizing scientific literature with retrieval-augmented lms, 2024. URL <https://arxiv.org/abs/2411.14199>.
- [5] Austin Baggio. autoresearch@home: Set up an AI research agent in 10 minutes. <https://ensue.dev/blog/autoresearch-at-home/>, 2026. Official project blog/documentation, accessed March 19, 2026.
- [6] Tong Bao, Mir Tafseer Nayeem, Davood Rafiei, and Chengzhi Zhang. SurveyGen: Quality-aware scientific survey generation with large language models. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 2025. URL <https://arxiv.org/abs/2508.17647>. EMNLP 2025 Main. arXiv:2508.17647. Anthology: <https://aclanthology.org/2025.emnlp-main.136/>.
- [7] Berkeley Sky Computing Lab. Skydiscover: Ai for scientific discovery. <https://sky.cs.berkeley.edu/project/skydiscover/>, 2026. Accessed: 2026-03-25.
- [8] Daniil A. Boiko, Robert MacKnight, Benjamin Kline, and Gabe Gomes. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578, 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-06792-0.
- [9] Bruce G. Buchanan and Edward A. Feigenbaum. Dendral and meta-dendral: Their applications dimension. *Artificial Intelligence*, 11(1–2):5–24, 1978. doi: 10.1016/0004-3702(78)90010-3. URL <https://www.sciencedirect.com/science/article/pii/0004370278900103>.
- [10] Eric Chamoun, Michael Schlichtkrull, and Andreas Vlachos. Automated focused feedback generation for scientific writing assistance. In *Findings of the Association for Computational Linguistics: ACL 2024*, 2024. URL <https://aclanthology.org/2024.findings-acl.580/>. Findings of ACL 2024. arXiv:2405.20477.
- [11] Yu Chao, Siyu Lin, Xiaorong Wang, Zhu Zhang, Zihan Zhou, Haoyu Wang, Shuo Wang, Jie Zhou, Zhiyuan Liu, and Maosong Sun. LLM × MapReduce-v3: Enabling interactive in-depth survey generation through a MCP-driven hierarchically modular agent system. In *Proceedings*

- of the 2025 Conference on Empirical Methods in Natural Language Processing: System Demonstrations, pages 688–695, 2025. URL <https://aclanthology.org/2025.emnlp-demos.51/>.
- [12] Jing Chen, Zhiheng Yang, Yixian Shen, Jie Liu, Adam Belloum, Paola Grosso, and Chrysa Papagianni. SurveyGen-I: Consistent scientific survey generation with evolving plans and memory-guided writing. *arXiv preprint arXiv:2508.14317*, 2025. doi: 10.48550/arXiv.2508.14317. URL <https://arxiv.org/abs/2508.14317>.
- [13] Ziru Chen, Shijie Chen, Yuting Ning, Qianheng Zhang, Boshi Wang, Botao Yu, Yifei Li, Zeyi Liao, Chen Wei, Zitong Lu, Vishal Dey, Mingyi Xue, Frazier N. Baker, Benjamin Burns, Daniel Adu-Ampratwum, Xuhui Huang, Xia Ning, Song Gao, Yu Su, and Huan Sun. Scienceagent-bench: Toward rigorous assessment of language agents for data-driven scientific discovery, 2025. URL <https://arxiv.org/abs/2410.05080>.
- [14] Mike D’Arcy, Tom Hope, Larry Birnbaum, and Doug Downey. MARG: Multi-agent review generation for scientific papers. *arXiv preprint arXiv:2401.04259*, 2024. doi: 10.48550/arXiv.2401.04259. URL <https://arxiv.org/abs/2401.04259>.
- [15] Yuanqi Du, Botao Yu, Tianyu Liu, Tony Shen, Junwu Chen, Jan G. Rittig, Kunyang Sun, Yikun Zhang, Zhangde Song, Bo Zhou, Cassandra Masschelein, Yingze Wang, Haorui Wang, Haojun Jia, Chao Zhang, Hongyu Zhao, Martin Ester, Teresa Head-Gordon, Carla P. Gomes, Huan Sun, Chenru Duan, Philippe Schwaller, and Wengong Jin. Accelerating scientific discovery with autonomous goal-evolving agents, 2025. URL <https://arxiv.org/abs/2512.21782>.
- [16] Brian D. Earp, Haotian Yuan, Julian Koplin, and Sebastian Porsdam Mann. The provenance problem: LLMs and the breakdown of citation norms, 2025. URL <https://arxiv.org/abs/2509.13365>. arXiv:2509.13365.
- [17] Elicit. Elicit: Ai for scientific research, 2026. *Elicit: AI for Scientific Research*. official product pages, accessed 2026. <https://elicit.com/welcome>.
- [18] Ensue. autoresearch@home. <https://www.ensue-network.ai/autoresearch>, 2026. Project page, accessed March 19, 2026.
- [19] Tony Feng, Trieu H. Trinh, Garrett Bingham, Dawsen Hwang, Yuri Chervonyi, Junehyuk Jung, Joonkyung Lee, Carlo Pagano, Sang hyun Kim, Federico Pasqualotto, Sergei Gukov, Jonathan N. Lee, Junsu Kim, Kaiying Hou, Golnaz Ghiasi, Yi Tay, YaGuang Li, Chenkai Kuang, Yuan Liu, Hanzhao Lin, Evan Zheran Liu, Nigamaa Nayakanti, Xiaomeng Yang, Heng-Tze Cheng, Demis Hassabis, Koray Kavukcuoglu, Quoc V. Le, and Thang Luong. Towards autonomous mathematics research. *arXiv preprint arXiv:2602.10177*, 2026. doi: 10.48550/arXiv.2602.10177.
- [20] Martin Funkquist, Ilia Kuznetsov, Yufang Hou, and Iryna Gurevych. CiteBench: A benchmark for scientific citation text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL <https://arxiv.org/abs/2212.09577>. EMNLP 2023. arXiv:2212.09577.

- [21] Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023. URL <https://arxiv.org/abs/2305.14627>. EMNLP 2023. arXiv:2305.14627.
- [22] Xian Gao, Jiacheng Ruan, Jingsheng Gao, Ting Liu, and Yuzhuo Fu. ReviewAgents: Bridging the gap between human and AI-generated paper reviews. *arXiv preprint arXiv:2503.08506*, 2025. doi: 10.48550/arXiv.2503.08506. URL <https://arxiv.org/abs/2503.08506>.
- [23] Xian Gao, Jiacheng Ruan, Zongyun Zhang, Qiyuan Liang, Guowei Wu, Shichao Song, Bingquan Liu, and Chengjie Sun. MMReview: A multidisciplinary and multimodal benchmark for LLM-based peer review automation, 2025. URL <https://arxiv.org/abs/2508.14146>. arXiv:2508.14146.
- [24] Gregory Hok Tjoan Go, Khang Ly, Anders Søgaaard, Amin Tabatabaei, Maarten de Rijke, and Xinyi Chen. Lira: A multi-agent framework for reliable and readable literature review generation, 2026. URL <https://arxiv.org/abs/2510.05138>.
- [25] Google DeepMind. Aletheia: A generator–verifier–reviser research agent powered by gemini deep think, 2026. *Aletheia: A Generator–Verifier–Reviser research agent powered by Gemini Deep Think*. Official blog, 2026. <https://deepmind.google/blog/accelerating-mathematical-and-scientific-discovery-with-gemini-deep-think/>.
- [26] Google Research and Google DeepMind. Accelerating scientific breakthroughs with an AI co-scientist. <https://research.google/blog/accelerating-scientific-breakthroughs-with-an-ai-co-scientist/>, 2025. Official research blog post.
- [27] Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025. doi: 10.1038/s41586-025-09422-z. URL <https://www.nature.com/articles/s41586-025-09422-z>.
- [28] Justin Hill and Hong Joo Ryoo. Grace: an agentic ai for particle physics experiment design and simulation, 2026. URL <https://arxiv.org/abs/2602.15039>.
- [29] Mengze Hong, Di Jiang, Chen Jason Zhang, Zhichao Duan, and Jianguo Zhang. CiteLLM: An agentic platform for trustworthy scientific reference discovery, 2026. URL <https://arxiv.org/abs/2602.23075>. arXiv:2602.23075.
- [30] Junyi Hou, Andre Lin Huikai, Nuo Chen, Yiwei Gong, and Bingsheng He. Paperdebugger: A plugin-based multi-agent system for in-editor academic writing, review, and editing. *arXiv preprint arXiv:2512.02589*, 2025. URL <https://arxiv.org/abs/2512.02589>.
- [31] Kexin Huang, Ying Jin, Ryan Li, Michael Y. Li, Emmanuel Candès, and Jure Leskovec. Automated hypothesis validation with agentic sequential falsifications, 2025. URL <https://arxiv.org/abs/2502.09858>.

- [32] Tal Ifargan, Lukas Hafner, Maor Kern, Ori Alcalay, and Roy Kishony. Autonomous LLM-driven research — from data to human-verifiable research papers. *NEJM AI*, 2(1), 2025. doi: 10.1056/AIoa2400555. URL <https://ai.nejm.org/doi/full/10.1056/AIoa2400555>.
- [33] Insilico Medicine. Dora community edition, 2026. *DORA Community Edition*. official release announcement and repository, 2026. Announcement: <https://insilico.com/news/kv5e80yc41-insilico-open-sources-dora-on-github-sup>; Repository: <https://github.com/insilicomedicine/DORA>.
- [34] InternAgent Team, Bo Zhang, Shiyang Feng, Xiangchao Yan, Jiakang Yuan, Runmin Ma, Yusong Hu, Zhiyin Yu, Xiaohan He, Songtao Huang, Shaowei Hou, Zheng Nie, Zhilong Wang, Jinyao Liu, Tianshuo Peng, Peng Ye, Dongzhan Zhou, Shufei Zhang, Xiaosong Wang, Yilan Zhang, Meng Li, Zhongying Tu, Xiangyu Yue, Wanli Ouyang, Bowen Zhou, and Lei Bai. InternAgent: When agent becomes the scientist – building closed-loop system from hypothesis to verification. *arXiv preprint arXiv:2505.16938*, 2025. doi: 10.48550/arXiv.2505.16938. URL <https://arxiv.org/abs/2505.16938>.
- [35] Intology AI. Zochi: Autonomous ai researcher achieves acl 2025 acceptance. <https://www.intology.ai/blog/zochi-acl>, 2025. Accessed: 2026-03-25.
- [36] Nilesh Jain, Rohit Yadav, and Andrej Karpathy. Bibby AI – AI latex editor writing assistant for researchers vs overleaf alternative vs OpenAI prism. *arXiv preprint arXiv:2602.16432*, 2026. doi: 10.48550/arXiv.2602.16432. URL <https://arxiv.org/abs/2602.16432>.
- [37] Yiqiao Jin, Qinlin Zhao, Yiyang Wang, Hao Chen, Kaijie Zhu, Yijia Xiao, and Jindong Wang. AgentReview: Exploring peer review dynamics with LLM agents. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, 2024. URL <https://arxiv.org/abs/2406.12708>. EMNLP 2024, Oral. arXiv:2406.12708. Repository: <https://github.com/ahren09/agentreview>.
- [38] Andrej Karpathy. autoresearch/program.md. <https://github.com/karpathy/autoresearch/blob/master/program.md>, 2026. Repository protocol file, accessed March 19, 2026.
- [39] Andrej Karpathy. karpathy/autoresearch: AI agents running research on single-GPU nanochat training automatically. <https://github.com/karpathy/autoresearch>, 2026. GitHub repository, accessed March 19, 2026.
- [40] Pat Langley. *Scientific discovery: Computational explorations of the creative processes*. MIT press, 1987.
- [41] Yutong Li, Lu Chen, Ayuan Liu, Yiding Wang, and Kai Yu. ChatCite: LLM agent with human workflow guidance for comparative literature summary. In *Proceedings of the 31st International Conference on Computational Linguistics (COLING)*, 2025. URL <https://arxiv.org/abs/2403.02574>. COLING 2025. arXiv:2403.02574.
- [42] Weixin Liang, Yuhui Zhang, Hancheng Cao, Binglu Wang, Daisy Ding, Xinyu Yang, Kailas Vodrahalli, Siyu He, Daniel Smith, Yian Yin, Daniel McFarland, and James Zou. Can large language models provide useful feedback on research papers? A large-scale empirical analysis. *NEJM AI*, 2024. URL <https://arxiv.org/abs/2310.01783>. arXiv:2310.01783.

- [43] Xun Liang, Jiawei Yang, Yezhaohui Wang, Chen Tang, Zifan Zheng, Shichao Song, Zehao Lin, Yebin Yang, Simin Niu, Hanyu Wang, Bo Tang, Feiyu Xiong, Keming Mao, and Zhiyu Li. Survey: Academic survey automation via large language models. *arXiv preprint arXiv:2502.14776*, 2025. URL <https://arxiv.org/abs/2502.14776>.
- [44] Jiaqi Liu, Shi Qiu, Peng Xia, Siwei Han, Letian Zhang, Guiming Chen, Haoqin Tu, Xinyu Yang, Jiawei Zhou, Hongtu Zhu, Yun Li, Yuyin Zhou, Zeyu Zheng, Cihang Xie, Mingyu Ding, and Huaxiu Yao. AutoRebuttalClaw: AI-powered academic rebuttal generation pipeline, 2026. URL <https://github.com/aiming-lab/AutoRebuttalClaw>.
- [45] Chris Lu, Cong Lu, Robert Tjarko Lange, Jakob Foerster, Jeff Clune, and David Ha. The AI scientist: Towards fully automated open-ended scientific discovery. *arXiv preprint arXiv:2408.06292*, 2024. doi: 10.48550/arXiv.2408.06292. URL <https://arxiv.org/abs/2408.06292>.
- [46] Chris Lu, Cong Lu, Robert Tjarko Lange, Yutaro Yamada, Shengran Hu, Jakob Foerster, David Ha, and Jeff Clune. Towards end-to-end automation of AI research. *Nature*, 651: 914–919, 2026. doi: 10.1038/s41586-026-10265-5. URL <https://www.nature.com/articles/s41586-026-10265-5>.
- [47] Indrajeet Mandal, Jitendra Soni, Mohd Zaki, Morten M. Smedskjaer, Katrin Wondraczek, Lothar Wondraczek, Nitya Nand Gosvami, and N. M. Anoop Krishnan. Evaluating large language model agents for automation of atomic force microscopy. *Nature Communications*, 16, 2025. URL <https://api.semanticscholar.org/CorpusID:282100257>.
- [48] Mutable State. `mutable-state-inc/autoresearch-at-home`. <https://github.com/mutable-state-inc/autoresearch-at-home>, 2026. GitHub repository, accessed March 19, 2026.
- [49] OpenAI. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024. doi: 10.48550/arXiv.2412.16720. URL <https://arxiv.org/abs/2412.16720>.
- [50] OpenAI. FrontierScience: Evaluating AI’s ability to perform scientific research tasks. OpenAI Technical Report, December 2025. URL <https://openai.com/index/frontierscience/>. Full paper available at <https://cdn.openai.com/pdf/2fcd284c-b468-4c21-8ee0-7a783933efcc/frontierscience-paper.pdf>.
- [51] Azim Ospanov, Farzan Farnia, and Roozbeh Yousefzadeh. Apollo: Automated llm and lean collaboration for advanced formal reasoning, 2025. URL <https://arxiv.org/abs/2505.05758>.
- [52] Overleaf. Ai features / ai assistant / error assist documentation, 2026. *AI Features / AI Assistant / Error Assist Documentation*. official documentation, accessed 2026. <https://docs.overleaf.com/integrations-and-add-ons/ai-features>.
- [53] Paperpal. Ai academic writing tool – comprehensive ai research assistant. <https://paperpal.com/>, 2026. Official product page.
- [54] PSI Inc. Psi: Building the future of scientific intelligence. <https://www.psi.inc>, 2026. Accessed: 2026-03-25.

- [55] Abhilasha Ravichander, Shruti Ghela, David Wadden, and Yejin Choi. HALoGEN: Fantastic LLM hallucinations and where to find them. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2025. URL <https://arxiv.org/abs/2501.08292>. ACL 2025 Long Papers. arXiv:2501.08292. Anthology: <https://aclanthology.org/2025.acl-long.71/>.
- [56] Yusuf Roohani, Andrew Lee, Qian Huang, Jian Vora, Zachary Steinhardt, Kexin Huang, Alexander Marson, Percy Liang, and Jure Leskovec. Biodiscoveryagent: An ai agent for designing genetic perturbation experiments, 2025. URL <https://arxiv.org/abs/2405.17631>.
- [57] Furkan Şahinuç, Ilia Kuznetsov, Yufang Hou, and Iryna Gurevych. Systematic task exploration with LLMs: A study in citation text generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024. URL <https://aclanthology.org/2024.acl-long.265/>. ACL 2024 Main. arXiv:2407.04046.
- [58] Samuel Schmidgall, Yusheng Su, Ze Wang, Ximeng Sun, Jialian Wu, Xiaodong Yu, Jiang Liu, Michael Moor, Zicheng Liu, and Emad Barsoum. Agent laboratory: Using LLM agents as research assistants. *arXiv preprint arXiv:2501.04227*, 2025. doi: 10.48550/arXiv.2501.04227. URL <https://arxiv.org/abs/2501.04227>.
- [59] Scholarcy. Article summarizer / literature review generator / integrations, 2026. *Article Summarizer / Literature Review Generator / Integrations*. official product pages, accessed 2026. <https://www.scholarcy.com/article-summarizer>.
- [60] Scholarly Document Processing Workshop Organizers. Scihal 2025 shared task: Hallucination detection for scientific assistants, 2025. URL <https://sdproc.org/2025/scihal.html>. Shared-task infrastructure for scientific hallucination analysis.
- [61] SciSpace. Ai writer / copilot for scientific research, 2026. *AI Writer / Copilot for Scientific Research*. official product pages, accessed 2026. <https://scispace.com/ai-writer>.
- [62] Michael D. Skarlinski, Sam Cox, Jon M. Laurent, James D. Braza, Michaela Hinks, Michael J. Hammerling, Manvitha Ponnappati, Samuel G. Rodrigues, and Andrew D. White. Language agents achieve superhuman synthesis of scientific knowledge, 2024. URL <https://arxiv.org/abs/2409.13740>.
- [63] Yiwen Song, Yale Song, Tomas Pfister, and Jinsung Yoon. Paperorchestra: A multi-agent framework for automated ai research paper writing, 2026. URL <https://arxiv.org/abs/2604.05018>.
- [64] Giulio Starace, Oliver Jaffe, Dane Sherburn, James Aung, Jun Shern Chan, Leon Maksin, Rachel Dias, Evan Mays, Benjamin Kinsella, Wyatt Thompson, Johannes Heidecke, Amelia Glaese, and Tejal Patwardhan. Paperbench: Evaluating ai’s ability to replicate ai research, 2025. URL <https://arxiv.org/abs/2504.01848>.
- [65] Zhaojun Sun, Xuzhou Zhu, Xuanhe Zhou, Xin Tong, Shuo Wang, Jie Fu, Guoliang Li, Zhiyuan Liu, and Fan Wu. SurveyBench: Can LLM(-agents) write academic surveys that align with reader needs? *arXiv preprint arXiv:2510.03120*, 2025. doi: 10.48550/arXiv.2510.03120. URL <https://arxiv.org/abs/2510.03120>.

- [66] Nicholas J. Szymanski, Brian Rendy, Yu Fei, Raghav E. Kumar, Tian He, David Milsted, Matthew J. McDermott, Matthew Gallant, Ekin D. Cubuk, Amil Merchant, Hyungjun Kim, Anubhav Jain, Christopher J. Bartel, Kristin Persson, Yu Zeng, and Gerbrand Ceder. An autonomous laboratory for the accelerated synthesis of inorganic materials. *Nature*, 624(7990): 86–91, 2023. ISSN 1476-4687. doi: 10.1038/s41586-023-06734-w.
- [67] Jiabin Tang, Lianghao Xia, Zhonghang Li, and Chao Huang. AI-researcher: Autonomous scientific innovation. *arXiv preprint arXiv:2505.18705*, 2025. doi: 10.48550/arXiv.2505.18705. URL <https://arxiv.org/abs/2505.18705>.
- [68] Nitya Thakkar, Mert Yuksekgonul, Jake Silberg, Animesh Garg, Nanyun Peng, Fei Sha, Rose Yu, Carl Vondrick, and James Zou. Can llm feedback enhance review quality? a randomized study of 20k reviews at iclr 2025, 2025. URL <https://arxiv.org/abs/2504.09737>.
- [69] Francisco Villaescusa-Navarro, Boris Bolliet, Pablo Villanueva-Domingo, Adrian E. Bayer, Aidan Acquah, Chetana Amancharla, Almog Barzilay-Siegal, Pablo Bermejo, Camille Bilodeau, Pablo Cárdenas Ramírez, Miles Cranmer, Urbano L. França, ChangHoon Hahn, Yan-Fei Jiang, Raul Jimenez, Jun-Young Lee, Antonio Lerario, Osman Mamun, Thomas Meier, Anupam A. Ojha, Pavlos Protopapas, Shimanto Roy, David N. Spergel, Pedro Tarancón-Álvarez, Ujjwal Tiwari, Matteo Viel, Digvijay Wadekar, Chi Wang, Bonny Y. Wang, Licong Xu, Yossi Yovel, Shuwen Yue, Wen-Han Zhou, Qiyao Zhu, Jiajun Zou, and Íñigo Zubeldia. The Denario project: Deep knowledge AI agents for scientific discovery. *arXiv preprint arXiv:2510.26887*, 2025. doi: 10.48550/arXiv.2510.26887. URL <https://arxiv.org/abs/2510.26887>.
- [70] David Waltz and Bruce G Buchanan. Automating science. *Science*, 324(5923):43–44, 2009.
- [71] Yidong Wang, Qi Guo, Wenjin Yao, Hongbo Zhang, Xin Zhang, Zhen Wu, Meishan Zhang, Xinyu Dai, Min Zhang, Qingsong Wen, Wei Ye, Shikun Zhang, and Yue Zhang. Autosurvey: Large language models can automatically write surveys. *arXiv preprint arXiv:2406.10252*, 2024. doi: 10.48550/arXiv.2406.10252. URL <https://arxiv.org/abs/2406.10252>.
- [72] Yubo Wang, Xueguang Ma, Ping Nie, Huaye Zeng, Zhiheng Lyu, Yuxuan Zhang, Benjamin Schneider, Yi Lu, Xiang Yue, and Wenhui Chen. ScholarCopilot: Training large language models for academic writing with accurate citations. *arXiv preprint arXiv:2504.00824*, 2025. doi: 10.48550/arXiv.2504.00824. URL <https://arxiv.org/abs/2504.00824>.
- [73] Zi Wang, Xingqiao Wang, Sangah Lee, and Xiaowei Xu. ARISE: Agentic rubric-guided iterative survey engine for automated scholarly paper generation. *arXiv preprint arXiv:2511.17689*, 2025. doi: 10.48550/arXiv.2511.17689. URL <https://arxiv.org/abs/2511.17689>.
- [74] Zifeng Wang, Benjamin Danek, and Jimeng Sun. Biodsa-1k: Benchmarking data science agents for biomedical research, 2025. URL <https://arxiv.org/abs/2505.16100>.
- [75] Lukas Weidener, Marko Brkić, Phillip Lee, Martin Karlsson, Kevin Noessler, and Paul Kohlhaas. From agent-only social networks to autonomous scientific research: Lessons from openclaw and moltbook, and the architecture of clawdlab and beach.science, 2026. URL <https://arxiv.org/abs/2602.19810>.

- [76] Yixuan Weng, Minjun Zhu, Guangsheng Bao, Hongbo Zhang, Jindong Wang, Yue Zhang, and Linyi Yang. CycleResearcher: Improving automated research via automated review. In *The Thirteenth International Conference on Learning Representations*, 2025. URL <https://openreview.net/forum?id=bjcsVLoHYs>. Poster.
- [77] Yixuan Weng, Minjun Zhu, Qiuji Xie, Qiyao Sun, Zhen Lin, Sifan Liu, and Yue Zhang. DeepScientist: Advancing frontier-pushing scientific findings progressively. *arXiv preprint arXiv:2509.26603*, 2025. doi: 10.48550/arXiv.2509.26603. URL <https://arxiv.org/abs/2509.26603>.
- [78] Writefull and Overleaf. Writefull for overleaf / texgpt. <https://help.writefull.com/writefull-for-overleaf--user-guide>, 2026. Official help and integration pages; see also <https://www.writefull.com/writefull-for-overleaf>.
- [79] Peng Xia, Jiaqi Liu, Shi Qiu, Siwei Han, Haoqin Tu, Xinyu Yang, Yuyin Zhou, Zeyu Zheng, Cihang Xie, and Huaxiu Yao. AutoResearchClaw: Fully autonomous and self-evolving research from idea to paper, 2026. URL <https://github.com/aiming-lab/AutoResearchClaw>.
- [80] Yihang Xiao, Jinyi Liu, Yan Zheng, Xiaohan Xie, Jianye Hao, Mingzhi Li, Ruitao Wang, Fei Ni, Yuxiao Li, Jintian Luo, Shaoqing Jiao, and Jiajie Peng. Cellagent: An llm-driven multi-agent framework for automated single-cell data analysis, 2024. URL <https://arxiv.org/abs/2407.09811>.
- [81] Yutaro Yamada, Robert Tjarko Lange, Cong Lu, Shengran Hu, Chris Lu, Jakob Foerster, Jeff Clune, and David Ha. The ai scientist-v2: Workshop-level automated scientific discovery via agentic tree search. *arXiv preprint arXiv:2504.08066*, 2025. doi: 10.48550/arXiv.2504.08066. URL <https://arxiv.org/abs/2504.08066>.
- [82] Xiangchao Yan, Shiyang Feng, Jiakang Yuan, Renqiu Xia, Bin Wang, Bo Zhang, and Lei Bai. Surveyforge: On the outline heuristics, memory-driven generation, and multi-dimensional evaluation for automated survey writing. *arXiv preprint arXiv:2503.04629*, 2025. URL <https://arxiv.org/abs/2503.04629>.
- [83] Haofei Yu, Keyang Xuan, Fenghai Li, Kunlun Zhu, Zijie Lei, Jiaxun Zhang, Ziheng Qi, Kyle Richardson, and Jiaxuan You. TinyScientist: An interactive, extensible, and controllable framework for building research agents. *arXiv preprint arXiv:2510.06579*, 2025. doi: 10.48550/arXiv.2510.06579. URL <https://arxiv.org/abs/2510.06579>.
- [84] Jiakang Yuan, Xiangchao Yan, Shiyang Feng, Bo Zhang, Tao Chen, Botian Shi, Wanli Ouyang, Yu Qiao, Lei Bai, and Bowen Zhou. Dolphin: Moving towards closed-loop auto-research through thinking, practice, and feedback. *arXiv preprint arXiv:2501.03916*, 2025. doi: 10.48550/arXiv.2501.03916. URL <https://arxiv.org/abs/2501.03916>.
- [85] Weizhe Yuan, Pengfei Liu, and Graham Neubig. Can we automate scientific reviewing? *Journal of Artificial Intelligence Research*, 75:171–212, 2022. URL <https://arxiv.org/abs/2102.00176>. JAIR Vol. 75. arXiv:2102.00176.

-
- [86] Minjun Zhu, Yixuan Weng, Linyi Yang, and Yue Zhang. DeepReview: Improving LLM-based paper review with human-like deep thinking process. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 29330–29355, 2025. doi: 10.18653/v1/2025.acl-long.1420. URL <https://aclanthology.org/2025.acl-long.1420/>.

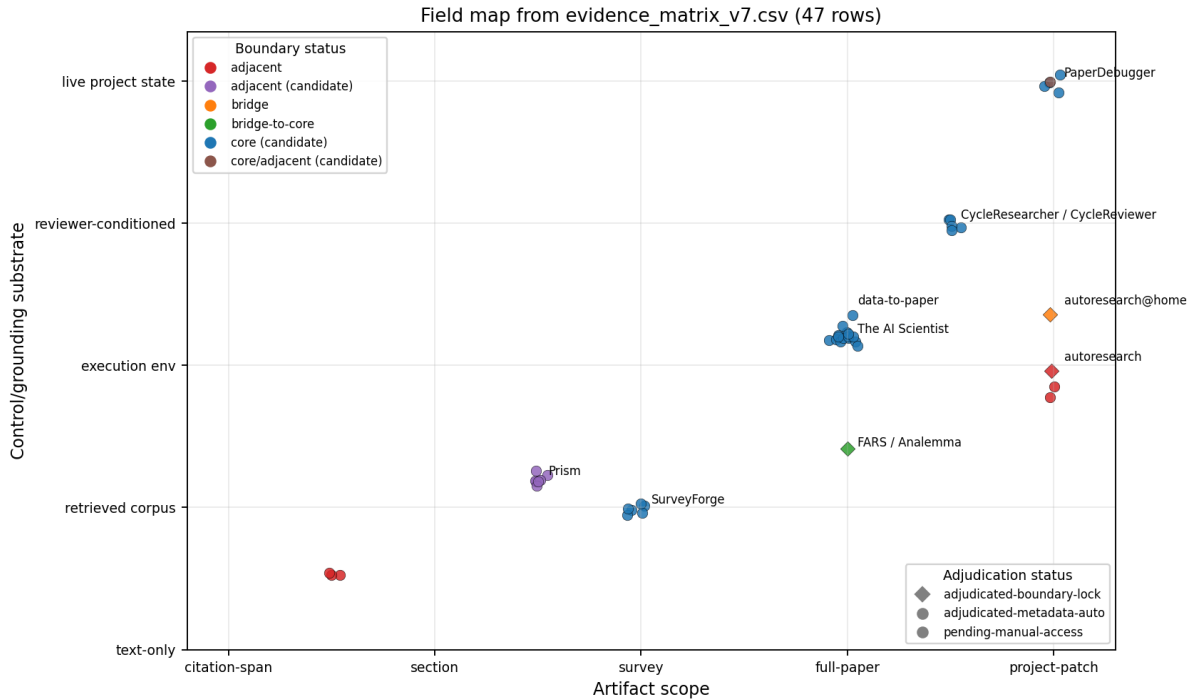


Figure 2. Field map rendered from evidence_matrix_v7.csv (47 rows). X-axis encodes artifact scope; Y-axis encodes grounding/control substrate; color indicates boundary status; marker shape indicates adjudication status class.

A Closed Products and Adjacent Copilots

This appendix collects systems that are relevant to the deployment landscape of scholarly writing, but not suitable as core comparison objects in the main survey. The reason is methodological rather than evaluative. Commercial and semi-commercial products such as OVERLEAF AI ASSIST, WRITEFULL, PAPERPAL, SCISPACE, ELICIT, and SCHOLARCY [17, 52, 53, 59, 61, 78] influence real-world academic workflows, sometimes at very large scale, yet do not expose enough architectural, training, or validation detail to support reproducible analysis under the framework used in this paper.

Their practical importance is therefore not in doubt, but their evidentiary status is different from that of the archival systems discussed in the main text. They are best understood as deployment-pattern evidence: editor-integrated drafting and revision tools, cross-platform writing assistants, and literature search or summarization products that shape how researchers actually work, without permitting strong claims about internal grounding or validation mechanisms. Including them in the main taxonomy on equal terms would collapse an important distinction between public scientific evidence and closed product behavior.

For that reason, these systems are kept in the appendix as boundary context rather than treated as technical baselines.