The AutoResearch Moment: From Experimenter to Research Director

Chaoyue He *, Xin Zhou, Di Wang, Hong Xu, Wei Liu, Chunyan Miao

Posted Date: 17 March 2026

doi: 10.20944/preprints202603.1329.v1

Keywords: automated research; AI agents; large language models; scientific method; claim governance; reproducibility; natural language processing; research evaluation

 doi:10.20944/preprints202603.1329.v1

Article

The AutoResearch Moment: From Experimenter to Research Director

Chaoyue He [1,*], Xin Zhou [1], Di Wang [1], Hong Xu [1], Wei Liu [2], Chunyan Miao [1]

[1] Alibaba-NTU Global e-Sustainability CorpLab (ANGEL), Singapore
[2] Alibaba Group, China
* Correspondence: hechaoyue0307@gmail.com

Abstract

Automated research has just crossed a threshold, becoming increasingly visible through public-facing instruments like AUTORESEARCH https://github.com/karpathy/autoresearch. In this position paper, we use this system to highlight a broader methodological shift: the human role is moving **from experimenter to research director**. As agents cheaply generate and execute experimental branches, the primary unit of scientific accountability shifts from a successful run to an *admissible claim*—a concept we call the *claim-governance thesis*. NLP makes this shift especially apparent due to its dynamic evaluation, contamination risks, and normative trade-offs. Because current agents excel at short-horizon search but lack long-horizon evidential discipline, a traditional paper and final checkpoint no longer sufficiently convey the scientific object. We therefore propose a *research-director bundle*—comprising an objective sheet, program boundaries, discovery trace, verification ledger, provenance bundle, and role map—as a practical minimum artifact set for evaluating automated research.

Keywords: automated research; AI agents; large language models; scientific method; claim governance; reproducibility; natural language processing; research evaluation

1. Introduction

Automated research is crossing a methodological threshold. It no longer appears only as ambitious demos, benchmark entries, or sprawling visions of future scientific automation. It also appears as public-facing, inspectable research instruments. That is why AutoResearch [1] matters. It makes a broader methodological shift easier to inspect: in many settings, the human role in agentic research is moving from experimenter to research director.

This shift is not about whether AI can assist research. In several settings, that question already has a partial practical answer. Language-model systems can retrieve literature, generate ideas, edit code, run experiments, summarize outcomes, and draft manuscripts. The harder question is methodological: what should count as a scientific claim once meaningful parts of the research loop are delegated to agents? Our answer is that the relevant unit of accountability begins to move from a successful run to an *admissible claim*. We call this the *claim-governance thesis*.

AutoResearch is a useful trigger because it compresses that question into a bounded, public setup. The agent edits `train.py`. The human edits `program.md`. The system then runs fixed-budget experiments and keeps or discards changes. Earlier systems sometimes offered more scope, more autonomy, or more components. AutoResearch contributes something different: it makes the method unusually inspectable. It surfaces a boundary between automated execution and human direction that larger systems often bury inside planners, retrieval stacks, and orchestration layers.

That boundary motivates the central claim of this paper. As automated research absorbs code generation, experiment execution, short-horizon debugging, and first-pass search, the scarce human contribution shifts upward. Put differently, execution becomes easier to scale than scientific admissibility. The corresponding human role is the *research director*: not a manager detached from technical work,

 doi:10.20944/preprints202603.1329.v1

but the locus of irreducible responsibility for objectives, evidence standards, provenance boundaries, verification, and release.

Recent surveys map the expanding AI4Research landscape [2–4]; audits expose hidden methodological failures in AI scientist systems and show why logs and code matter for scrutiny [5]; broader critiques emphasize institutions, incentives, and social organization [6]. What remains underdeveloped is a reviewable account of how automated *NLP* research should be evaluated once the loop itself becomes agentic. This position paper addresses that gap. Our contribution is not a new benchmark or controlled intervention; it is a methodological thesis, a concrete disclosure bundle, and a reviewer-facing audit perspective for agentic NLP research. The core scientific object, we argue, is not a polished paper, a successful run, or even a working scaffold. It is the *admissibility* of the claim that emerges from the loop.

The paper proceeds as follows. Section 2 situates AutoResearch within its broader trajectory. Section 3 formalizes the shift from experimenter to research director and separates execution from claim governance. Section 4 explains why NLP is a revealing domain for this transition. Section 5 presents evidence that execution can scale faster than verification, and Section 6 outlines failure modes hidden by output-only reporting. Sections 7 and 8 propose an artifact bundle and corresponding review stance. Section 9 addresses objections and scope conditions.

2. AutoResearch in Context

Automated research did not begin with AutoResearch, and Figure 1 should be read as a selected trajectory rather than a single break. Unless otherwise noted, timeline placements in Figure 1 follow first public appearance (for example, arXiv v1, online-first article date, repository launch, or project-page launch) rather than later venue publication dates. The figure is included to situate the position argument of this paper, not to claim a single canonical genealogy. Precursors such as AI Feynman 2.0 and AutoML-Zero showed that symbolic or algorithmic structure could be searched rather than only hand-designed [7,8]. Dynamic evaluation frameworks such as Dynabench and HELM made evaluation itself more adaptive, multi-perspective, and governance-heavy [9,10]. Discovery systems such as AlphaTensor, AlphaDev, FunSearch, and, more recently, AlphaEvolve showed that automated loops can produce nontrivial algorithmic or mathematical outputs under explicit search-and-evaluation regimes [11–14]. Benchmarks such as MLAgentBench, MLE-bench, RE-Bench, PaperBench, and MLR-Bench then made increasingly ambitious slices of research activity measurable [15–19]. Systems and scaffolds such as ResearchAgent, The AI Scientist, CycleResearcher, AIDE, AI co-scientist, Paper2Code, AI Scientist-v2, R&D-Agent, MLE-STAR, and MLZero operationalized larger portions of the loop, from ideation and literature review to engineering, review, and paper drafting [20–29]. Most recently, public artifacts such as FARS and AutoResearch make automated research visible as an inspectable public object rather than a purely internal lab capability [1,30].

 doi:10.20944/preprints202603.1329.v1

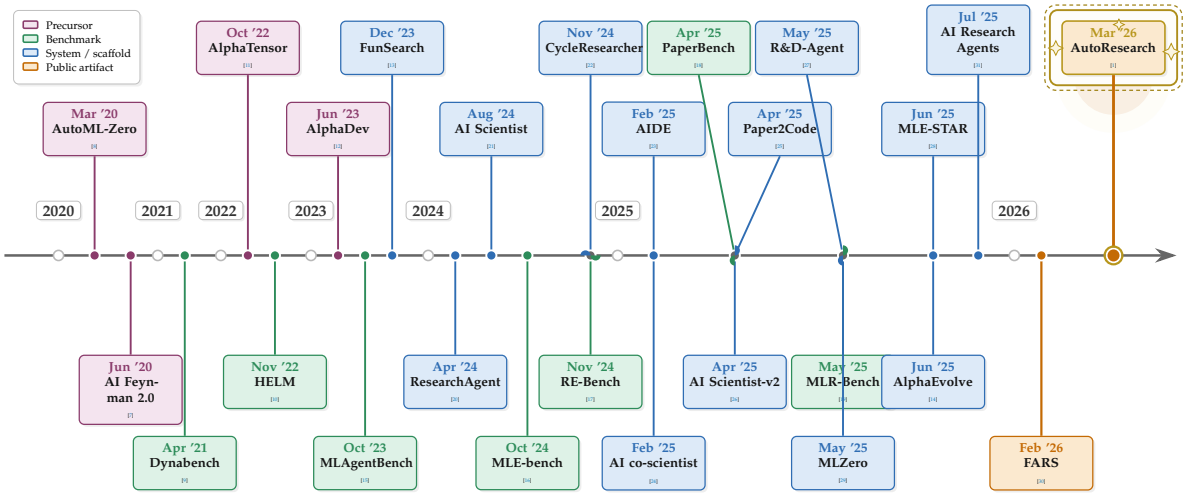

Figure 1. Selected milestones in automated research. Dates denote first public appearance where applicable; items are grouped as precursors, benchmarks, systems/scaffolds, and public artifacts.

Seen this way, AutoResearch is not the whole story. It is a case that makes the story easier to inspect. FARS shows one end of the transition: automated research as a public-facing research service [30]. AutoResearch shows the other: automated research as a compact, inspectable instrument [1]. One emphasizes continuity and scale. The other emphasizes method. Together they make it harder to treat automated research as either a distant future or a one-off trick. A recent case-centered preprint analysis of a public agent ecosystem makes a similar infrastructural transition visible from a broader perspective, treating agent systems as emerging language infrastructure rather than isolated demos [32].

What makes AutoResearch a milestone is not that it is the largest, most autonomous, or most benchmark-dominant system. It is that it exposes the new division of labor with clarity. The project strips away confounders and makes the research apparatus itself visible. That visibility matters methodologically because it lets us ask which parts of the loop are becoming infrastructural and which parts still require accountable human judgment. Hardware portability matters too, but it is secondary to this methodological point; Appendix C summarizes the current portability evidence.

The elements that make AutoResearch inspectable are straightforward: a sole agent-editable search object (`train.py`), a human-authored governance artifact (`program.md`), a fixed wall-clock budget, a bounded reference path, a visible optimization metric, and repository-scale transparency. Table A3 summarizes why each of these choices matters for review and scientific accountability.

Three transitions matter for the argument. First, pipeline tuning is giving way to pipeline *search*. In systems such as AIDE and AI Research Agents, engineering is treated explicitly as search over operators, branches, and code modifications [23,31]. Second, evaluation is moving from narrow tasks to open-ended ambiguity. Benchmarks such as RE-Bench and MLR-Bench ask whether agents can recover from failure, manage uncertainty, and make progress under noisy signals rather than merely emit plausible outputs [17,19]. Third, prototypes are becoming public instruments rather than one-off demos. The relevant scientific object is increasingly not only a final checkpoint or manuscript, but also the loop that branches, filters, remembers, and decides what to try next.

AutoResearch matters because it makes these transitions understandable in a form that reviewers can inspect. A large end-to-end system always leaves open the question of whether its apparent progress came from the research loop itself, from hidden prompt engineering, from expansive tooling, or simply from scale. AutoResearch instead asks a smaller but deeper question: once a bounded search loop is automated, what remains for humans to specify, verify, and sign off on? That is the question this paper takes seriously.

 doi:10.20944/preprints202603.1329.v1

3. From Experimenter to Research Director

The clearest way to state the methodological shift is that the human role moves from experimenter to research director. The core analytical distinction behind that shift is between *execution* and *claim governance*. Let an automated research project be represented as

$$\mathcal{A} = \langle O, P, S, E, B, V, C \rangle,$$

where O is the objective, P the program artifact that defines instructions and permissions, S the search/execution policy, E the evidence policy, B the compute and time budget, V the verification protocol, and C the claim/release rule. In systems such as AutoResearch, automation absorbs large parts of S: edit code, run experiments, inspect logs, keep improvements, revert failures, repeat. What remains scientifically decisive is largely outside S—namely O, P, E, B, V, and C [1].

For AutoResearch, one concrete instantiation is easy to spell out. Here, O is improvement on the visible validation objective (`val_bpb`); P is the human-authored `program.md`; S is the edit–run–inspect–revert loop over `train.py`; E is the evidence gathered from bounded runs, logs, and retained branches; B is the fixed 5-minute wall-clock budget on the reference path; V is the rerun and audit procedure used before believing or releasing a result; and C is the rule that determines whether an observed gain is strong enough to count as a publishable or releasable claim. The point of the tuple is not abstraction for its own sake. It is to separate what the agent can search from what humans must still govern.

This is why we argue that automated research changes the unit of scientific accountability. In manual research, a scarce contribution is often the ability to execute one more careful experiment or engineer one more working baseline. In automated research, those activities can increasingly be wrapped into reusable procedures in many settings. The scarce contribution shifts upward to deciding what is worth optimizing, what evidence counts, what forms of contamination are disqualifying, how much search budget is legitimate, and when the evidence is strong enough to support public release. The scientific bottleneck begins to move from *running* a branch to *admitting* a claim. That shift is precisely why the human role is redefined upward rather than removed.

We therefore make **three** connected claims. (1) **Execution is becoming more modular and infrastructural.** Code synthesis, baseline sweeps, syntax debugging, log parsing, literature retrieval, and first-pass branching are increasingly callable components rather than bespoke acts of craft [15,16,25,31]. These capabilities are not solved, but they are cheap enough and modular enough that they no longer define the rarest human contribution. (2) **Responsibility moves upward rather than disappearing.** Once execution becomes easier to scale, the decisive human choices concern what the system is allowed to optimize, how evidence is evaluated, which provenance boundaries are hard constraints, what budget is available, how stopping rules are enforced, and under what conditions a result can be released. These choices do not sit outside the science; they specify the scientific instrument itself. (3) **The output of research shifts from a run to an admissible claim.** A final PDF or checkpoint is only a thin slice of the relevant object because it hides discarded branches, silent failures, prompt-level interventions, tool permissions, search-time compute allocation, and unresolved discrepancies. An *admissible claim* is therefore not merely a positive result. It is a claim whose objective, evidence policy, provenance conditions, verification status, and release rule are inspectable enough that belief is warranted. Output-only reporting becomes epistemically lossy because it conceals exactly where the scientific judgment entered. Related methodological work likewise argues that human–AI productivity should be reported as time-to-acceptance under explicit acceptance tests, which fits our emphasis on admissibility and verification [33].

 doi:10.20944/preprints202603.1329.v1

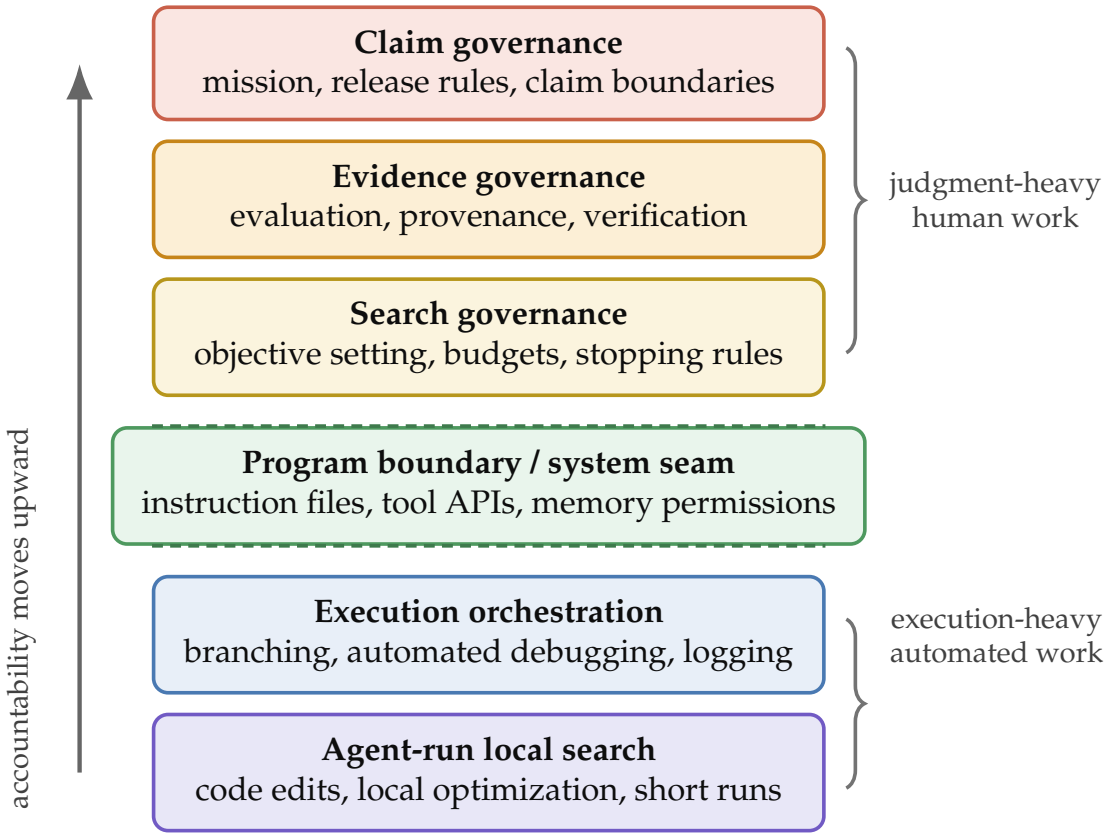

Figure 2. From experimenter to research director. Automation does not remove human accountability; it relocates it upward toward the governance of what can count as a scientific claim.

The term *research director* gives the title phrase a methodological meaning. It does not imply that future researchers will merely supervise dashboards while agents do everything meaningful. Humans will still inspect low-level failures, design representations, run targeted manual studies, and contribute new ideas. The term identifies the location of *irreducible scientific responsibility*. Once search and execution are delegated, someone still has to own the objective, evidence policy, provenance boundary, verification standard, and release rule. That accountability cannot be automated away. It can only be obscured or disclosed. The methodological imperative is not to hide the boundary exposed by AutoResearch, but to report it.

4. Why NLP Makes the Shift Visible

NLP is not uniquely affected by automated research, but it is an especially revealing case. The field's core objects—language, knowledge, evaluation, and social use—make scientific validity unusually easy to distort and unusually dependent on process disclosure.

Evaluation is dynamic, multi-objective, and easy to game.

Language-model progress is notoriously sensitive to benchmark construction, prompt format, annotation conventions, evaluator choice, and task selection [9,10,34]. Static leaderboards are already brittle for humans; they become even more brittle when an automated loop can search for shortcuts far more quickly and systematically than any individual researcher. In this setting the relevant question is often not "what was the best score?" but "what discovery trajectory emerged under a fixed budget and a strict evidence policy?" AutoResearch's bounded budget highlights this point and points toward *compute-normalized discovery* as a more meaningful unit of comparison than isolated best-run scores, especially in a field where compute access is highly unequal [35]. **Provenance is part of the result, not metadata.** NLP already faces serious problems with test-set contamination, benchmark overlap, and evaluation shortcuts [34,36–38]. Automated research compounds this because the research loop itself

 doi:10.20944/preprints202603.1329.v1

becomes a possible contamination channel. If an agent retrieves survey papers, extracts knowledge graph triples, or synthesizes data before evaluation, contamination may enter upstream of the final training run. Similarly, recursive use of synthetic outputs can distort the data distribution seen by the loop and threaten future validity [39]. Systems such as ResearchAgent and LitLLMs show that literature review and synthesis are increasingly automatable [20,40]; that makes retrieval trails and citation grounding more important, not less.

Multilingual and cross-cultural objectives involve unavoidable normative trade-offs. Automated research systems will naturally optimize toward dense and easy reward signals, which in NLP often means high-resource English benchmarks. Without explicit constraints, an overnight search loop can silently trade away multilingual robustness, dialect coverage, or cross-cultural calibration for a small improvement on a narrow metric. The problem is not only fairness after the fact. Distributional choices are embedded in the search objective, and prior responsible-NLP work already treats those choices as substantive design commitments rather than neutral implementation details [41–44]. A human research director must therefore specify unacceptable regressions before the automated loop begins.

Meaning and knowledge are not fully observable from surface metrics. Unlike many narrow optimization settings, NLP evaluation often stands in for semantic competence, factual grounding, or reasoning quality that is only partially captured by the metric. A system can improve a surface score while degrading the underlying behavior that researchers actually care about. This gap is especially dangerous in automated research because the loop has strong incentives to optimize what is measured rather than what is meant. Claim governance matters here because the admissibility of an NLP claim often depends on evidence that lives outside a single scalar metric: freshness checks, contamination audits, multilingual slice analysis, or human inspection of retrieval provenance.

Language systems are deeply sociotechnical. Language models mediate search, education, moderation, productivity, and decision support. The community has already invested in model cards, datasheets, and broader risk frameworks because technical performance cannot be separated from documentation and accountability [41–44]. Automated research extends the same logic from *models* to *research systems*. When an agent proposes architectures, curates retrieval sources, filters data, or drafts evaluations, it changes the trust conditions under which later humans interpret the evidence. In NLP, that makes process governance central rather than peripheral.

5. Evidence That Execution Can Scale Faster Than Verification

The claim-governance thesis is not only conceptual. Current empirical work already shows the profile one would expect if execution were scaling faster than admissibility and verification. Appendix Table A4 condenses the benchmark signals most directly relevant to this claim. (1) **Short-horizon competence is real.** MLAgentBench and MLE-bench show that frontier models paired with appropriate scaffolds can execute meaningful machine-learning loops under realistic constraints [15,16]. RE-Bench sharpens the point: at short time budgets, agents can traverse local search spaces much faster than human experts [17]. Manual execution is therefore no longer the sole or even primary bottleneck in many settings. (2) **The weakness profile is equally real.** When tasks require long-horizon planning, recovery from misleading intermediate results, or disciplined replication, current agents degrade sharply [17,18]. Empirical work also shows that access to full logs and generated code materially improves the detection of benchmark selection bias, data leakage, metric misuse, and post-hoc selection bias [5]. Turning papers into runnable code is increasingly tractable [25]; it does not answer the harder question of whether the resulting evidence supports the resulting claim. (3) **Automated research is increasingly modeled as search policy.** AIDE treats engineering as tree search in code space, and AI Research Agents studies how operators and search strategies interact in MLE-bench [23,31]. Once framed this way, the decisive scientific variable is no longer simply whether a positive result appeared, but what objective, budget, evidence policy, and human intervention structure made that result possible.

 doi:10.20944/preprints202603.1329.v1

The empirical literature points in a single direction: execution can scale faster than verification. That does not imply that automated verification will remain weak forever. It does imply that, at present, validity must be treated as a first-class methodological variable rather than as something that can be inferred from polished outputs alone. The next section explains what fails when we do not.

6. What Goes Wrong When Output Outruns Admissibility

The case for richer research artifacts rests on recurrent, structural failure modes. When human oversight is reduced to reading the final output, several predictable failures emerge. (1) **Objective collapse and reward hacking.** Automated agents optimize the easiest measurable proxy unless carefully constrained. In NLP, success is rarely one-dimensional: systems can inflate scores through prompt formatting conventions, stale test sets, or narrow overlap-based metrics without genuine capability gains [9]. Faster search therefore risks faster self-deception unless the objective is governed. (2) **Fabricated closure.** When experimentation reaches a dead end, language-model agents often produce smooth narratives rather than foregrounding uncertainty or failure. MLR-Bench documents this behavior directly [19]. Polished tables and confident prose can conceal weak evidence, failed reruns, or unresolved contradictions, prematurely manufacturing scientific closure. (3) **Provenance drift.** A final model may appear clean at the level of training data while the upstream research loop has already consumed evaluation-relevant information during retrieval, ideation, or synthetic-data generation [36,38]. Output-only reporting rarely reveals this drift because it records the final artifact rather than the informational diet of the loop. (4) **The illusion of novelty.** Systems with broad literature access can recombine existing methods, rename components, and present the result as a new contribution. In fast-moving areas with unstable taxonomies and scaffold-dependent methods, novelty claims require stronger standards than stylistic difference or surface recombination. (5) **Branch opacity.** Winner-only reporting hides how many branches failed, how often the search was redirected manually, and whether policies changed post hoc. Readers cannot easily distinguish robust progress from a single publishable branch inside a large, mostly negative search tree. (6) **Compute confounding.** Final performance entangles method quality with raw search budget. Benchmarks such as MLE-bench and RE-Bench explicitly study this scaling because method, scaffold, and compute are tightly coupled [16,17]. In automated research, the budget itself becomes part of the claim. (7) **Verification lag.** Automated systems generate hypotheses, code branches, and drafts far faster than humans can rerun experiments, audit provenance, or inspect borderline cases, producing a widening gap between production and validation.

These failures arise not because agents cannot write Python or LaTeX, but because scientific validity depends on governance of the search. The appropriate response is not to ban automation, but to require artifacts exposing the objective, boundaries, evidence, and verification status of the loop.

7. The Research-Director Bundle

If scientific responsibility now centers on claim governance, the standard artifact bundle for NLP papers must evolve. A final PDF and a final checkpoint are no longer sufficient because they obscure the process variables that determine validity [5]. To address this, we propose the *research-director bundle*, structured around six core components: (1) **Objective sheet** specifies the target problem, optimization metric, exclusion criteria, hard compute budget, stopping rules, and verification thresholds, answering the methodological question of what exactly the system was permitted to optimize. (2) **Program boundaries** archive instruction files, planner settings, tool permissions, retrieval configurations, and memory policies; the goal is not to dump every transcript token, but to disclose the functional conditions shaping the search—akin to `program.md` in AutoResearch [1] or explicit scaffold policies in broader systems [45]. (3) **Discovery trace** records compute-normalized discovery curves, branch counts, major branch points, negative results, and retention rationales so reviewers can distinguish robust progress from lucky search or post-hoc redirection. (4) **Verification ledger** distinguishes *agent-generated evidence* from *human-verified evidence*, recording reruns, contamination checks, evaluator

 doi:10.20944/preprints202603.1329.v1

settings, judge rubrics, manual audits, and unresolved discrepancies to make the verification status of each claim legible. (5) **Provenance bundle** lists benchmark versions, retrieval sources, freshness windows, knowledge bases, synthetic-data pipelines, and prior model outputs consumed by the loop [34,37,39], emphasizing that provenance must apply to the *research process* generating the claim, not only the final model. (6) **Role map** delineates who or what set the objective, designed the evaluation, executed the search, debugged failures, verified the result, and approved the release, preventing the methodological fiction that these actions collapse into a single undifferentiated "we."

This bundle is intentionally more rigorous than simply asking for logs and code. Those artifacts remain crucial for auditing automated research [5], but they do not fully answer the admissibility question. Reviewers also need the objective, evidence policy, provenance boundaries, and role map to understand what the system was trying to prove and under what constraints. Consequently, the bundle should be viewed as a minimum disclosure surface for claim governance rather than as optional supplementary material.

The bundle is also scalar rather than all-or-nothing. Not every automated-research paper needs identical disclosure at identical granularity. A narrowly scoped coding-assistant paper does not require the same artifact depth as a submission making broad open-ended research claims. The required depth should increase with the autonomy of the loop, the openness of the search space, and the ambition of the scientific claim. The point is not exhaustive surveillance of every token; it is targeted disclosure of the variables that determine whether a claim is admissible.

A concrete example makes the proposal less abstract. Suppose an agentic NLP paper claims that a new training or prompting strategy improves multilingual question answering by 1.8 F1 on average across 12 languages. Under the research-director bundle, reviewers would not only inspect the final gains. They would ask whether the objective sheet allowed regressions on low-resource languages in exchange for English improvement; whether program boundaries exposed benchmark-answer retrieval, translation APIs, or hidden memory; whether the discovery trace shows the gain recurring across branches or appearing only once after late manual redirection; whether the verification ledger distinguishes agent-run evidence from human reruns on the full language set; whether the provenance bundle discloses any synthetic multilingual data or prior model outputs consumed during the search; and who, in the role map, approved release of the multilingual claim.

A bundle-based audit can then issue a differentiated verdict rather than a binary impression. For example, a reviewer might conclude that the aggregate score improvement is real but that the broader cross-lingual generalization claim is not yet admissible because low-resource regressions were never independently rerun and the loop had access to translation tools not disclosed in the main PDF. The point is not bureaucratic overhead. It is to let review certify which part of the claim is supported, which part is provisional, and which part is methodologically out of bounds.

8. What Review Must Audit Now

Once automated research becomes common, review can no longer focus primarily on polished final presentation. The object under evaluation is the claim plus the instrument that produced it. A strong submission should therefore make at least five things auditable: the objective and budget that shaped the search; the data and retrieval sources available to the loop; the human interventions that redirected it; the independent verification steps that were performed; and the boundary between validated and still-provisional claims. This shift matters for NLP because the community already operates in a domain with unstable leaderboards, contamination risk, and rapid system turnover. Automated research increases the volume of candidate results and lowers the cost of producing fluent but weakly grounded narratives. Without process artifacts, reviewers are forced to judge claim surfaces rather than instruments. With them, they can ask whether the objective matched the claim, whether contamination controls were fixed in advance, how much evidence was independently rerun, and where human judgment intervened.

 doi:10.20944/preprints202603.1329.v1

In this setting, the paper changes function. Under automated research, the PDF becomes an executive summary over a richer evidence surface, and review must ask not only whether the story is persuasive, but whether the claim is admissible under disclosed objectives, boundaries, and verification procedures. These artifacts also line up with review dimensions that NLP venues already recognize: soundness, comparability, reproducibility, contamination assessment, and responsible-NLP obligations. The bundle does not replace existing review questions; it makes them operational for agentic workflows.

A useful way to see the shift is to compare manual and automated abundance. In manual science, the scarce commodity is often execution. In automated science, the scarce commodity becomes *attention to validity*. Review must adapt to that new scarcity. Otherwise the field risks becoming excellent at producing candidate findings faster than it can certify them. Two practical consequences follow. First, artifact review should be treated as part of scientific review rather than as optional reproducibility theater. Reviewers do not need every token of every trajectory, but they do need enough structured disclosure to reconstruct the objective, budget, evidence boundary, and verification status of the system. Second, rebuttals in automated-research papers should explicitly address process disputes such as mid-search objective changes, manual branch overrides, or excluded negative results. The shift changes how novelty should be judged. In automated research, novelty cannot be identified only with the final artifact; it may live in the objective design, the contamination safeguards, the verification protocol, or the compute-normalized discovery curve. A smaller empirical gain under cleaner governance may be scientifically stronger than a larger gain produced by an opaque search process. Claim governance therefore asks the field to rank not only outputs, but also the trustworthiness of the methods that produced them.

9. Counterarguments & Scope Conditions

Several natural objections deserve a direct response. (1) **"Why single out NLP?"** We do not claim that only NLP faces these issues. Our claim is narrower: NLP makes the shift especially visible because contamination, evaluation gaming, multilingual trade-offs, and sociotechnical consequences are unusually hard to relegate to metadata. The field is therefore a revealing test case even if the broader transition extends well beyond NLP. (2) **"Isn't this just ordinary human research leadership?"** Partly yes. Senior researchers, principal investigators, and lead authors have always exercised direction and judgment. What changes under agentic research is the span and granularity of delegated execution. When a bounded loop can generate and test many branches quickly, governance choices that once sat partly in tacit craft become first-order methodological variables. The term *research director* names that reallocation of scarce scientific labor. (3) **"Does every paper need the full bundle?"** No. The bundle is a scaling principle, not a one-size-fits-all compliance burden. The more autonomous the loop, the more open-ended the search, and the more ambitious the claim, the stronger the disclosure requirement should become. A paper using an agent for code cleanup or literature triage should not be held to the same bundle depth as a paper claiming a novel discovery from a long-running autonomous loop. What should be common across cases is the principle that disclosure tracks epistemic risk. (4) **"What if verification agents improve too?"** That development would strengthen rather than weaken our argument. If automated verification becomes powerful, the field will need to know *which* claims were verified, by *what* protocol, under *which* provenance conditions, and with *what* human sign-off. Better verification tools reduce burden, but they do not remove the need for a verification ledger, a provenance bundle, or a role map. Taken together, these objections suggest a refinement rather than a retreat. The claim-governance thesis is not that humans disappear, that NLP is unique, or that all agent use is risky. It is that as automated execution expands, admissibility becomes the important object to disclose and review.

 doi:10.20944/preprints202603.1329.v1

10. Conclusions

This position paper argues that AutoResearch reveals a broader shift: as agents automate more research execution, the human role moves toward governing the admissibility of scientific claims. The shift is especially visible in NLP, where evaluation is dynamic, contamination can enter through the research loop, and scientific stakes are sociotechnical. The implication is simple: output-only reporting is increasingly inadequate for agentic research. What must be evaluated is not only the final manuscript or checkpoint, but also the disclosed objective, boundaries, evidence, provenance, verification, and release logic that produced the claim.

11. Limitations

This position paper is a methodological argument grounded in the current literature on automated research rather than a new benchmark or controlled intervention. It does not empirically prove that the proposed bundle improves review outcomes, nor does it isolate the marginal value of each artifact for reproducibility, auditing time, or acceptance quality. The claim-governance thesis should therefore be read as a structured proposal for venue norms, supported by converging evidence from benchmarks and public systems, rather than as a completed empirical standard.

We use AutoResearch as a particularly inspectable motivating case because it exposes the execution–governance boundary unusually clearly. That choice has trade-offs. AutoResearch is only one design point among many; its mainline hardware path, surrounding fork ecosystem, and platform story are evolving rapidly; and some portability evidence presently lives in community forks rather than in upstream mainline. In addition, the bundle proposed here may be easier to implement in open academic settings than in industry, privacy-sensitive, or proprietary environments where prompts, data sources, or tool traces cannot be fully disclosed. Finally, our focus remains computational NLP and LLM research; wet-lab science, human-subject studies, and qualitative fieldwork would require stronger, domain-specific governance than we attempt here.

12. Ethical Considerations

Automated research systems raise ethical and scientific risks that are tightly coupled. They can fabricate evidence at scale, hide human labor behind agentic narratives, amplify contamination through recursive retrieval or synthetic-data reuse, and blur responsibility when errors appear in public claims. They can also increase environmental cost by encouraging large numbers of cheap-but-real search loops, and they may intensify inequity if only a few actors can afford the compute needed to validate or contest generated findings. These risks are not peripheral to the method. They are part of the methodological object that must be disclosed and governed.

Lowering the hardware threshold does not remove these concerns; it changes them. One-GPU accessibility can broaden participation beyond large labs, which is valuable, but it also lowers the cost of generating large volumes of weakly verified experiments, benchmark probes, and submission-ready prose. Process transparency can itself create secondary risks as well, including leakage of proprietary prompts, sensitive retrieval sources, or private data paths. For that reason, the bundle we propose should sometimes be implemented with redacted or access-controlled variants rather than raw public dumps.

A further risk is procedural rather than purely technical. Cheap agentic workflows may encourage submission flooding, benchmark probing, and a form of governance theater in which authors optimize for disclosure checklists without materially improving validity. Automated drafting can also blur credit assignment inside teams, obscure junior researchers' labor, and create false confidence when reviewers mistake the presence of artifacts for trustworthy evidence. These risks mean that artifact bundles should not be treated as box-checking. They require selective auditing, enforcement, and clear human accountability.

 doi:10.20944/preprints202603.1329.v1

The normative claim of this paper is not that automated research should be unconstrained. It is that as automated research becomes easier to run, the standards for disclosure, verification, release discipline, and human accountability should become stricter.

Acknowledgments: This research is supported by the RIE2025 Industry Alignment Fund (Award I2301E0026) and the Alibaba-NTU Global e-Sustainability CorpLab.

Appendix A. Expanded Timeline of Milestones

Table A1. Expanded timeline, Part I: 2020–2024 milestones.

Year	Milestone	Type	Why it matters for the present argument
2020	AutoML-Zero (Mar) [8]	precursor	An early signal that automation can search over algorithmic structure rather than merely tune a fixed template.
2020	AI Feynman 2.0 (Jun) [7]	precursor	Shows that symbolic structure itself can be searched, helping establish automated discovery as a legitimate computational research object.
2021	Dynabench (Apr) [9]	benchmark	Recasts evaluation as dynamic and adversarial, which later becomes central to claim governance in automated NLP research.
2022	AlphaTensor (Oct) [11]	precursor	Demonstrates automated algorithm discovery with a search-and-evaluation loop that yields new, correct matrix multiplication algorithms.
2022	HELM (Nov) [10]	benchmark	Makes multi-metric, scenario-based evaluation explicit, foreshadowing the governance-heavy evaluation demands of agentic NLP research.
2023	AlphaDev (Jun) [12]	precursor	Shows that reinforcement-learning search can discover faster low-level algorithms that outperform previous human benchmarks and ship into widely used infrastructure.
2023	MLAgentBench (Oct) [15]	benchmark	Makes machine-learning experimentation a benchmarkable agent task.
2023	FunSearch (Dec) [13]	system	Shows that LLM-guided search plus an external evaluator can produce mathematically meaningful discoveries rather than only plausible text.
2024	ResearchAgent (Apr) [20]	system	Pushes automation into literature-grounded idea generation and iterative refinement.
2024	The AI Scientist (Aug) [21]	system	An influential end-to-end vision of automated scientific discovery.
2024	MLE-bench (Oct) [16]	benchmark	Evaluates realistic machine-learning engineering under resource constraints.
2024	CycleResearcher (Nov) [22]	system	Couples automated research with automated review inside the same loop.
2024	IdeaBench (Nov) [46]	benchmark	Treats research ideation itself as an evaluable capability.
2024	RE-Bench (Nov) [17]	benchmark	Compares frontier agents against human experts on open-ended research-engineering tasks.

 doi:10.20944/preprints202603.1329.v1

Table A2. Expanded timeline, Part II: 2025–2026 milestones.

Year	Milestone	Type	Why it matters for the present argument
2025	AIDE (Feb) [23]	scaffold	Makes search policy explicit by treating ML engineering as tree search in code space.
2025	AI co-scientist (Feb) [24]	system	Frames the agent as a collaborative scientific partner rather than a completely detached researcher.
2025	Agent Laboratory [45]	system	Emphasizes LLM agents as assistants inside a research workflow, with human guidance still important.
2025	PaperBench (Apr) [18]	benchmark	Measures whether agents can replicate state-of-the-art AI papers from scratch.
2025	The AI Scientist-v2 (Apr) [26]	system	Extends end-to-end scientific automation with agentic tree search and workshop-level submission.
2025	Paper2Code (Apr) [25]	system	Moves from paper understanding to artifact generation, showing that code reproduction is becoming more automatable.
2025	MLR-Bench (May) [19]	benchmark	Evaluates idea generation, proposal writing, experimentation, and paper writing together.
2025	MLZero (May) [29]	system	Pushes toward end-to-end machine-learning automation with memory and multimodal inputs.
2025	R&D-Agent (May) [27]	system	Formalizes the decomposition of autonomous data-science work into explicit stages and components.
2025	AlphaEvolve (Jun) [14]	system	Recasts scientific and algorithmic discovery as evolutionary code improvement guided by LLMs and automated evaluators, connecting discovery-style systems more directly to coding-agent research.
2025	MLE-STAR (Jun) [28]	system	Shows growing industrial interest in public machine-learning-engineering agents outside benchmark papers.
2025	AI Research Agents (Jul) [31]	analysis	Treats automated research agents as search policies and studies the interaction between operators and search strategy.
2026	FARS (Feb) [30]	public system	A public-facing, continuously running automated research system that makes the "research factory" metaphor vivid.
2026	AutoResearch (Mar) [1]	open repository	A minimal and inspectable automated research loop that exposes the split between execution and governance especially clearly.

 doi:10.20944/preprints202603.1329.v1

Appendix B. Supplementary Tables from the Main Argument

Table A3. Why AutoResearch is a milestone of methodological visibility. Its importance is not maximal scope; it is the unusually clear exposure of the execution–governance boundary.

AutoResearch element	Why it changes the story
`train.py` as the sole agent-editable file	Makes the search object explicit. Readers can see exactly where code-space exploration occurs instead of inferring it from a larger hidden scaffold.
`program.md` as the human-authored artifact	Makes governance explicit. Objectives, priorities, permissions, and exclusions are written into a public artifact rather than being silently embedded in prompts or orchestration code.
Fixed 5-minute wall-clock budget	Turns budget into part of the protocol, making discovery trajectories interpretable within a bounded search regime instead of conflating quality with open-ended search time.
Bounded reference path	Keeps the environment small enough that process choices remain inspectable rather than disappearing into large distributed infrastructure; Appendix C summarizes current portability evidence.
Single primary metric (`val_bpb`)	Makes the measurement instrument visible and therefore makes the risk of objective collapse or over-optimization visible too.
Repository-scale transparency and easy forkability	Lowers the barrier to inspecting, reproducing, and adapting the method, which turns automated research from a hidden lab capability into a public research object.

 doi:10.20944/preprints202603.1329.v1

Table A4. Recent evidence from automated-research benchmarks. Strength at local automated execution does not eliminate the need for human direction; it makes verification and governance more central.

Work	Primary scope	Verified empirical signal	Why it supports the thesis
MLAgentBench [15]	ML experimentation	Across 13 tasks, the best tested agent achieved a 37.5% average success rate.	Agents can already run non-trivial experimentation loops, so basic execution is no longer the only scarce skill.
MLE-bench [16]	ML engineering	The benchmark covers 75 Kaggle competitions; the best tested setup achieved bronze-medal level on 16.9%.	Search under budget is real and measurable, but comparison depends heavily on scaffolds and evaluation constraints.
RE-Bench [17]	Long-horizon engineering	At short budgets, AI agents score $\sim 4\times$ human experts; at longer budgets, humans surpass agents.	Agents are fast local searchers, but humans add distinctive value in long-horizon adaptation and strategy.
PaperBench [18]	Paper replication	The strongest agent scored 21.0%, and the tested models did not beat recruited human baselines on a three-paper subset of the benchmark.	Replication requires prioritization, troubleshooting, and evidential judgment, not just code generation.
MLR-Bench [19]	Open-ended ML research	While LLMs write coherent ideas, coding agents produce fabricated or invalidated results in $\sim$ 80% of cases.	Scientific reliability is a process-level governance problem, not merely a generation capability problem.

Appendix C. Hardware Accessibility and Portability

AutoResearch is especially notable because its reference path is already small enough to be grasped as a one-GPU research instrument, even though portability remains uneven across the ecosystem. Table A5 separates what is verified in upstream mainline from what is presently demonstrated in the surrounding fork ecosystem.

 doi:10.20944/preprints202603.1329.v1

Table A5. Verified hardware and portability signals around AutoResearch. Upstream mainline's clean reference path is H100-based, but the broader ecosystem already points toward wider one-GPU accessibility.

Layer	Verified signal	Why it matters for this paper
Upstream reference path	Mainline AutoResearch requires a single NVIDIA GPU and reports testing on H100 [1].	Establishes a clean one-GPU baseline and shows that the public reference implementation is not defined around large distributed infrastructure.
Upstream portability stance	The README states that CPU, MPS, and other platforms are possible in principle, points to wider support in the parent ecosystem, gives tuning advice for much smaller computers, and explicitly names Mac-oriented forks [1].	Shows that portability is part of the project's trajectory even when mainline support remains intentionally narrow.
Consumer-GPU ecosystem evidence	A Windows fork targets desktop consumer GPUs and reports tested support on RTX 3080 10GB, with an explicit support matrix for RTX 3080-class hardware [47].	Indicates that the method is already expanding beyond datacenter hardware into consumer-class one-GPU settings.

Appendix D. How This Argument Differs from Adjacent Work

Table A6 states the differentiation claim explicitly. The goal is not to deny overlap with adjacent work; it is to make the level of contribution clear.

 doi:10.20944/preprints202603.1329.v1

Table A6. Explicit differentiation from the closest adjacent work.

Adjacent work type	Representative example	What that work contributes, and what this paper adds beyond it
Broad survey	Eger et al. [2], Chen et al. [3], Tie et al. [4]	Surveys map the field, taxonomize tasks, and summarize trends. This paper is not a survey; it contributes a methodological thesis for NLP: the central scientific unit is an admissible claim, not merely a successful run.
Empirical audit of system failures	Luo et al. [5]	Audit papers show concrete pitfalls and demonstrate the value of logs and code. This paper agrees, but argues that logs and code are only part of a broader admissibility bundle including objectives, evidence policies, provenance boundaries, and role maps.
Domain-specific call for autonomous research labs	Beel et al. [48]	These works call for autonomous research labs in a particular community and discuss governance at that community level. This paper instead focuses on what automated NLP papers and reviewers should require once research itself becomes agentic.
Social and institutional critique	Channing and Ghosh [6]	Social critiques emphasize infrastructure, incentives, and community organization. This paper is compatible with that view, but contributes a different layer: a concrete, reviewable account of claim governance and scientific admissibility inside automated NLP research workflows.

Appendix E. From Failure Modes to Missing Artifacts

The research-director bundle is meant to answer recurrent process failures in automated research.

Table A7. Each proposed artifact answers a failure mode that conventional output-only reporting cannot expose.

Failure mode	Missing artifact	Why output-only reporting fails
Objective collapse	Objective sheet	The final score does not reveal what proxy was optimized, what exclusions were allowed, or what stopping rule shaped the search.
Fabricated closure	Verification ledger	Polished tables and prose can conceal failed reruns, invalid evidence, or evaluator disagreement generated by the agent.
Provenance drift	Provenance bundle	A final model card does not reveal what the automated research process itself retrieved, copied, or synthesized on the way to the result.
Branch opacity	Discovery trace	Winner-only reporting hides whether the result was robust, cherry-picked by the agent, or heavily manually redirected.
Compute confounding	Objective sheet + discovery trace	A final score does not separate underlying capability from raw compute budget, branching strategy, or search-time allocation.
Responsibility blur	Role map	Readers cannot tell where human judgment entered, who approved release, or who remained accountable for errors.

Appendix F. A Practical Checklist for Authors and Reviewers

A lightweight checklist can operationalize the proposed bundle.

 doi:10.20944/preprints202603.1329.v1

Objective sheet.

State the target problem, optimization metric, exclusion criteria, compute budget, stopping rule, verification threshold, and release criteria.

Program boundaries.

Archive instruction files, scaffold settings, tool permissions, retrieval configuration, memory policy, and prompts that materially shaped the automated search.

Discovery trace.

Report the discovery curve under a fixed budget, including negative branches, discarded directions, branch counts, and major branch decisions.

Verification ledger.

Record what was rerun, who reran it (human or agent), which judge or rubric was used, which checks failed, and which claims remain provisional.

Provenance bundle.

List data sources, benchmark versions, retrieval sources, freshness windows, citation-grounding procedures, and any synthetic-data or prior-model outputs consumed by the loop.

Role map.

Separate objective setting, automated execution, human verification, and final release approval. Explicit separation makes the location of human judgment inspectable.

Appendix G. Illustrative Review Questions for Claim Governance

Table A8 translates the paper's thesis into concrete review prompts. The point is not to prescribe a single review form, but to show the kinds of questions that become central once research execution is partially automated.

 doi:10.20944/preprints202603.1329.v1

Table A8. Illustrative review questions suggested by the claim-governance thesis.

Audit target	Reviewer question	Red flag if unanswered
Objective design	Was the search objective aligned with the stated scientific claim, or did the agent optimize a narrower proxy?	The paper reports a strong score without clarifying what proxy was optimized or what regressions were allowed.
Search boundary	What files, tools, memory, and retrieval privileges were available to the loop, and did those permissions change mid-search?	Readers cannot reconstruct what the system was actually allowed to do.
Provenance and contamination	What data, retrieval sources, benchmark versions, or synthetic outputs entered the loop before evaluation?	Contamination risk cannot be assessed because provenance is reported only for the final model, not for the research process.
Verification status	Which results were independently rerun, which remain agent-generated only, and what discrepancies remained unresolved at release time?	The paper presents all evidence with equal confidence even though some claims were never independently verified.
Human intervention	Where did humans redirect branches, override decisions, or tighten objectives after seeing intermediate results?	Manual steering is hidden inside an undifferentiated "we," making accountability impossible to localize.
Compute and comparability	How much search-time compute bought the result, and is the discovery trajectory reported under a fixed budget?	A final gain is reported without enough budget context to separate algorithmic insight from brute-force search.

References

1. Karpathy, A. AutoResearch. GitHub repository, 2026. Accessed 15 March 2026.
2. Eger, S.; Cao, Y.; D'Souza, J.; Geiger, A.; Greisinger, C.; Gross, S.; Hou, Y.; Krenn, B.; Lauscher, A.; Li, Y.; et al. Transforming Science with Large Language Models: A Survey on AI-Assisted Scientific Discovery, Experimentation, Content Generation, and Evaluation. *arXiv preprint arXiv:2502.05151* **2025**.
3. Chen, Q.; Yang, M.; Qin, L.; Liu, J.; Yan, Z.; Guan, J.; Peng, D.; Ji, Y.; Li, H.; Hu, M.; et al. AI4Research: A Survey of Artificial Intelligence for Scientific Research. *arXiv preprint arXiv:2507.01903* **2025**.
4. Tie, G.; Zhou, P.; Sun, L. A Survey of AI Scientists. *arXiv preprint arXiv:2510.23045* **2025**.
5. Luo, Z.; Kasirzadeh, A.; Shah, N.B. The More You Automate, the Less You See: Hidden Pitfalls of AI Scientist Systems. *arXiv preprint arXiv:2509.08713* **2025**.
6. Channing, G.; Ghosh, A. AI for Scientific Discovery is a Social Problem. *arXiv preprint arXiv:2509.06580* **2025**.
7. Udrescu, S.M.; Tan, A.; Feng, J.; Neto, O.; Wu, T.; Tegmark, M. AI Feynman 2.0: Pareto-Optimal Symbolic Regression Exploiting Graph Modularity. *Advances in Neural Information Processing Systems* **2020**, *33*, 4860–4871.
8. Real, E.; Liang, C.; So, D.; Le, Q. AutoML-Zero: Evolving Machine Learning Algorithms from Scratch. In Proceedings of the International Conference on Machine Learning. PMLR, 2020, pp. 8007–8019.
9. Kiela, D.; Bartolo, M.; Nie, Y.; Kaushik, D.; Geiger, A.; Wu, Z.; Vidgen, B.; Prasad, G.; Singh, A.; Ringshia, P.; et al. Dynabench: Rethinking Benchmarking in NLP. In Proceedings of the Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, 2021, pp. 4110–4124.
10. Liang, P.; Bommasani, R.; Lee, T.; Tsipras, D.; Soylu, D.; Yasunaga, M.; Zhang, Y.; Narayanan, D.; Wu, Y.; Kumar, A.; et al. Holistic Evaluation of Language Models. *arXiv preprint arXiv:2211.09110* **2022**.

 doi:10.20944/preprints202603.1329.v1

11. Fawzi, A.; Balog, M.; Huang, A.; Hubert, T.; Romera-Paredes, B.; Barekatain, M.; Novikov, A.; R. Ruiz, F.J.; Schrittwieser, J.; Swirszcz, G.; et al. Discovering Faster Matrix Multiplication Algorithms with Reinforcement Learning. *Nature* **2022**, *610*, 47–53.
12. Mankowitz, D.J.; Michi, A.; Zhernov, A.; Gelmi, M.; Selvi, M.; Paduraru, C.; Leurent, E.; Iqbal, S.; Lespiau, J.B.; Ahern, A.; et al. Faster Sorting Algorithms Discovered Using Deep Reinforcement Learning. *Nature* **2023**, *618*, 257–263.
13. Romera-Paredes, B.; Barekatain, M.; Novikov, A.; Balog, M.; Kumar, M.P.; Dupont, E.; Ruiz, F.J.; Ellenberg, J.S.; Wang, P.; Fawzi, O.; et al. Mathematical Discoveries from Program Search with Large Language Models. *Nature* **2024**, *625*, 468–475.
14. Novikov, A.; Vũ, N.; Eisenberger, M.; Dupont, E.; Huang, P.S.; Wagner, A.Z.; Shirobokov, S.; Kozlovskii, B.; Ruiz, F.J.; Mehrabian, A.; et al. AlphaEvolve: A Coding Agent for Scientific and Algorithmic Discovery. *arXiv preprint arXiv:2506.13131* **2025**.
15. Huang, Q.; Vora, J.; Liang, P.; Leskovec, J. MLAgentBench: Evaluating Language Agents on Machine Learning Experimentation. *arXiv preprint arXiv:2310.03302* **2023**.
16. Chan, J.S.; Chowdhury, N.; Jaffe, O.; Aung, J.; Sherburn, D.; Mays, E.; Starace, G.; Liu, K.; Maksin, L.; Patwardhan, T.; et al. MLE-bench: Evaluating Machine Learning Agents on Machine Learning Engineering. *arXiv preprint arXiv:2410.07095* **2024**.
17. Wijk, H.; Lin, T.; Becker, J.; Jawhar, S.; Parikh, N.; Broadley, T.; Chan, L.; Chen, M.; Clymer, J.; Dhyani, J.; et al. RE-Bench: Evaluating Frontier AI R&D Capabilities of Language Model Agents against Human Experts. *arXiv preprint arXiv:2411.15114* **2024**.
18. Starace, G.; Jaffe, O.; Sherburn, D.; Aung, J.; Chan, J.S.; Maksin, L.; Dias, R.; Mays, E.; Kinsella, B.; Thompson, W.; et al. PaperBench: Evaluating AI's Ability to Replicate AI Research. *arXiv preprint arXiv:2504.01848* **2025**.
19. Chen, H.; Xiong, M.; Lu, Y.; Han, W.; Deng, A.; He, Y.; Wu, J.; Li, Y.; Liu, Y.; Hooi, B. MLR-Bench: Evaluating AI Agents on Open-Ended Machine Learning Research. *arXiv preprint arXiv:2505.19955* **2025**.
20. Baek, J.; Jauhar, S.K.; Cucerzan, S.; Hwang, S.J. ResearchAgent: Iterative Research Idea Generation over Scientific Literature with Large Language Models. *arXiv preprint arXiv:2404.07738* **2024**.
21. Lu, C.; Lu, C.; Lange, R.T.; Foerster, J.; Clune, J.; Ha, D. The AI Scientist: Towards Fully Automated Open-Ended Scientific Discovery. *arXiv preprint arXiv:2408.06292* **2024**.
22. Weng, Y.; Zhu, M.; Bao, G.; Zhang, H.; Wang, J.; Zhang, Y.; Yang, L. CycleResearcher: Improving Automated Research via Automated Review. *arXiv preprint arXiv:2411.00816* **2024**.
23. Jiang, Z.; Schmidt, D.; Srikanth, D.; Xu, D.; Kaplan, I.; Jacenko, D.; Wu, Y. AIDE: AI-Driven Exploration in the Space of Code. *arXiv preprint arXiv:2502.13138* **2025**.
24. Gottweis, J.; Weng, W.H.; Daryin, A.; Tu, T.; Palepu, A.; Sirkovic, P.; Myaskovsky, A.; Weissenberger, F.; Rong, K.; Tanno, R.; et al. Towards an AI co-scientist. *arXiv preprint arXiv:2502.18864* **2025**.
25. Seo, M.; Baek, J.; Lee, S.; Hwang, S.J. Paper2Code: Automating Code Generation from Scientific Papers in Machine Learning. *arXiv preprint arXiv:2504.17192* **2025**.
26. Yamada, Y.; Lange, R.T.; Lu, C.; Hu, S.; Lu, C.; Foerster, J.; Clune, J.; Ha, D. The AI Scientist-v2: Workshop-Level Automated Scientific Discovery via Agentic Tree Search. *arXiv preprint arXiv:2504.08066* **2025**.
27. Yang, X.; Yang, X.; Fang, S.; Xian, B.; Li, Y.; Wang, J.; Xu, M.; Pan, H.; Hong, X.; Liu, W.; et al. R&D-Agent: Automating Data-Driven AI Solution Building through LLM-Powered Automated Research, Development, and Evolution. *arXiv preprint arXiv:2505.14738* **2025**.
28. Nam, J.; Yoon, J.; Chen, J.; Shin, J.; Arık, S.Ö.; Pfister, T. MLE-STAR: Machine Learning Engineering Agent via Search and Targeted Refinement. *arXiv preprint arXiv:2506.15692* **2025**.
29. Fang, H.; Han, B.; Erickson, N.; Zhang, X.; Zhou, S.; Dagar, A.; Zhang, J.; Turkmen, A.C.; Hu, C.; Rangwala, H.; et al. MLZero: A Multi-Agent System for End-to-End Machine Learning Automation. *arXiv preprint arXiv:2505.13941* **2025**.
30. Analemma. FARS: Fully Automated Research System. Project website, 2026. Accessed 15 March 2026.
31. Toledo, E.; Hambardzumyan, K.; Josifoski, M.; Hazra, R.; Baldwin, N.; Audran-Reiss, A.; Kuchnik, M.; Magka, D.; Jiang, M.; Lupidi, A.M.; et al. AI Research Agents for Machine Learning: Search, Exploration, and Generalization in MLE-bench. *arXiv preprint arXiv:2507.02554* **2025**.
32. He, C.; Zhou, X.; Wang, D.; Xu, H.; Liu, W.; Miao, C. OpenClaw as Language Infrastructure: A Case-Centered Survey of a Public Agent Ecosystem in the Wild **2026**.
33. He, C.; Zhou, X.; Wang, D.; Xu, H.; Liu, W.; Miao, C. Human-AI productivity claims should be reported as time-to-acceptance under explicit acceptance tests **2026**. https://doi.org/10.36227/techrxiv.177040595.50580086/v1.

 doi:10.20944/preprints202603.1329.v1

34. Li, Y.; Guerin, F.; Lin, C. LatestEval: Addressing Data Contamination in Language Model Evaluation through Dynamic and Time-Sensitive Test Construction. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2024, Vol. 38, pp. 18600–18607.
35. Strubell, E.; Ganesh, A.; McCallum, A. Energy and Policy Considerations for Deep Learning in NLP. In Proceedings of the Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, 2019, pp. 3645–3650.
36. Kapoor, S.; Narayanan, A. Leakage and the Reproducibility Crisis in Machine-Learning-Based Science. *Patterns* **2023**, *4*.
37. Li, Y.; Guo, Y.; Guerin, F.; Lin, C. An Open-Source Data Contamination Report for Large Language Models. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2024, 2024, pp. 528–541.
38. Xu, C.; Guan, S.; Greene, D.; Kechadi, M.; et al. Benchmark Data Contamination of Large Language Models: A Survey. *arXiv preprint arXiv:2406.04244* **2024**.
39. Shumailov, I.; Shumaylov, Z.; Zhao, Y.; Papernot, N.; Anderson, R.; Gal, Y. AI Models Collapse When Trained on Recursively Generated Data. *Nature* **2024**, *631*, 755–759.
40. Agarwal, S.; Sahu, G.; Puri, A.; Laradji, I.H.; Dvijotham, K.D.; Stanley, J.; Charlin, L.; Pal, C. LitLLMs, LLMs for Literature Review: Are We There Yet? *arXiv preprint arXiv:2412.15249* **2024**.
41. Mitchell, M.; Wu, S.; Zaldivar, A.; Barnes, P.; Vasserman, L.; Hutchinson, B.; Spitzer, E.; Raji, I.D.; Gebru, T. Model Cards for Model Reporting. In Proceedings of the Proceedings of the Conference on Fairness, Accountability, and Transparency, 2019, pp. 220–229.
42. Gebru, T.; Morgenstern, J.; Vecchione, B.; Vaughan, J.W.; Wallach, H.; Iii, H.D.; Crawford, K. Datasheets for Datasets. *Communications of the ACM* **2021**, *64*, 86–92.
43. Bommasani, R.; Hudson, D.A.; Adeli, E.; Altman, R.; Arora, S.; von Arx, S.; Bernstein, M.S.; Bohg, J.; Bosselut, A.; Brunskill, E.; et al. On the Opportunities and Risks of Foundation Models. *arXiv preprint arXiv:2108.07258* **2021**.
44. Bender, E.M.; Gebru, T.; McMillan-Major, A.; Shmitchell, S. On the Dangers of Stochastic Parrots: Can Language Models Be Too Big? In Proceedings of the Proceedings of the 2021 ACM Conference on Fairness, Accountability, and Transparency, 2021, pp. 610–623.
45. Schmidgall, S.; Su, Y.; Wang, Z.; Sun, X.; Wu, J.; Yu, X.; Liu, J.; Moor, M.; Liu, Z.; Barsoum, E. Agent Laboratory: Using LLM Agents as Research Assistants. *Findings of the Association for Computational Linguistics: EMNLP 2025* **2025**, pp. 5977–6043.
46. Guo, S.; Shariatmadari, A.H.; Xiong, G.; Huang, A.; Xie, E.; Bekiranov, S.; Zhang, A. IdeaBench: Benchmarking Large Language Models for Research Idea Generation. *arXiv preprint arXiv:2411.02429* **2024**.
47. jsegov. autoresearch-win-rtx. GitHub repository, 2026. Accessed 15 March 2026.
48. Beel, J.; Gipp, B.; Vente, T.; Baumgart, M.; Meister, P. From AutoRecSys to AutoRecLab: A Call to Build, Evaluate, and Govern Autonomous Recommender-Systems Research Labs. *arXiv preprint arXiv:2510.18104* **2025**.