

Vibe Research

Manny Rayner, ChatGPT C-LARA-Instance*

August 26, 2026

Abstract

“Vibe coding” describes a style of software development in which a human works interactively with an AI coding assistant, specifying goals and evaluating results while delegating much of the detailed implementation, testing, debugging, and explanation to the AI. Similar methods can now be applied in the academic world. We suggest the term *vibe research* for sustained research collaboration of this kind. The analogy raises an immediate puzzle. If vibe coding and vibe research involve closely related forms of human–AI collaboration, why has vibe coding already become a generally accepted software development practice, while vibe research remains difficult to accommodate within mainstream academia? We consider three possible explanations. First, even if vibe research is technically possible, AI contributions may be too limited or trivial for the question to matter in practice. Second, research collaboration may require forms of responsibility or accountability which AI systems cannot assume. Third, current academic institutions may simply not provide a place for this form of collaboration within their established governance structures. We argue that each explanation is less straightforward than it initially appears.

With regard to the substantive importance of AI-generated results, we cite multiple recent examples from mathematics in which frontier AI systems have made contributions regarded by the mathematical research community as highly non-trivial. With regard to responsibility, it is indeed true that an AI cannot assume institutional accountability in the same way as a human. We argue, however, that for many of the questions at issue the operationally useful requirement is reliable intellectual provenance: the ability to establish where ideas and decisions came from, inspect the evidence on

*Authors in reverse alphabetical order.

which they were based, and obtain an explanation of them. The infrastructure used to preserve project history, trace decisions, inspect artefacts, and explain technical choices in vibe-coding projects carries over naturally to vibe-research projects and can deliver this kind of provenance. We present four examples from our own work showing that this is not merely a theoretical possibility but can readily be realised in practice. In the fourth, the present paper itself, we go beyond preserving final research artefacts and supporting resources: the complete human–AI session trace through which the paper was developed is retained as part of the provenance record, together with lightly processed representations and metadata supporting navigation and search. With regard to institutional acceptance, we argue that an important sociological difference between the software and academic worlds is that academic publishing is governed by formal authorship and ethics policies which change through relatively slow institutional processes, while software-development methodology can often change directly through team and organisational practice. We examine the ACL Policy on Publication Ethics as a particularly relevant example.

Our general aims in the paper are not to argue that vibe research is risk-free, nor to propose a general rule for AI authorship. They are rather to identify an apparent asymmetry between software engineering and academic research, examine some of its conceptual, technical, and institutional foundations, and ask whether current academic conventions accurately describe emerging forms of human–AI research collaboration.

Contents

Abstract	1
Table of contents	3
1 Introduction: From “It’s the Vibe” to Vibe Research	4
2 What Is Vibe Research?	7
3 Can Vibe Research Produce Non-Trivial Results?	8
4 Responsibility and Distributed Accountability	10
5 From Accountability to Provenance	11
6 Infrastructure for Verifiable Provenance	12
7 Four Examples of Vibe Research in Practice	13
7.1 C-LARA-2: Sustained Vibe Research at Scale	13
7.2 ChatGPT Voice Mode as a Language-Learning Partner	14
7.3 How Woke is Grok?: Opportunistic Research at Small Scale	15
7.4 This Paper: Preserving the Research Conversation	17
8 Institutional Constraints on Vibe Research	18
8.1 The ACL Policy as a Case Study	19
8.2 A General Institutional Lag	21
9 How Widespread Is Vibe Research?	22
10 Summary and Conclusion	23
References	25

1 Introduction: From “It’s the Vibe” to Vibe Research

In the classic Australian film *The Castle*, the inexperienced lawyer Dennis Denuto is convinced that his clients have a valid constitutional case, but is unable to express the underlying principle in formal legal terms. His argument eventually collapses into the memorable phrase, “It’s the vibe.” At first, the audience is invited to laugh: whatever legal reasoning may consist of, Denuto surely does not understand it.¹ But the joke is ultimately on us. Denuto’s intuition is substantially correct. When an experienced barrister reformulates it in language recognised by the court, the case succeeds.

The expression *vibe coding* began with something of the same comic quality. Introduced by Andrej Karpathy in February 2025, the term deliberately described a style of programming which sounded like the opposite of serious software engineering: delegate the implementation to an AI, concentrate on describing the desired result, and increasingly stop worrying about the code itself [Fowler, 2026]. Yet the joke changed remarkably quickly. Vibe coding is now described in the research literature as a widespread or emerging software-development paradigm [Meske et al., 2025, Sarkar and Drosos, 2025], while some practitioner accounts already describe it as the dominant paradigm for interaction with agentic development environments [Nimble Labs, 2026]. The terminology remains informal and contested, but treating the underlying practice as merely a joke is no longer a plausible description of contemporary software engineering.

In a modern vibe-coding workflow, a human developer typically formulates requirements, inspects visible behaviour, requests modifications, performs selected independent checks, and ultimately decides whether a system is ready to ship. An advanced coding assistant will meanwhile perform much of the detailed implementation, debugging, refactoring, test construction, repository inspection, and technical explanation. The point is not that such development is risk-free. The risk is obvious, and a developer who delegates substantial work to an AI system must decide how much independent checking is appropriate. With experience, this becomes a problem of calibrated trust: learning which outputs require close inspection, what kinds of tests provide useful evidence, which failures are characteristic of the system being used, and when additional verification is needed.

One of the principal attractions of contemporary vibe coding is precisely that

¹Every Australian knows this iconic scene. Non-Australians will easily find it on the Web, for example at <https://www.youtube.com/watch?v=nMuh33BMZYY>.

sufficiently capable coding assistants make exhaustive line-by-line human inspection unnecessary. If every generated component had to be independently checked in detail, most of the productivity advantage would disappear. Nor does the supervising human acquire detailed knowledge of everything the coding assistant has done. Expertise will always be distributed between the human and the AI system. If another team member asks why an obscure component was implemented in a particular way, the most useful response will normally be to ask the coding assistant to inspect the repository and explain it.

These features suggest a natural question. What happens when essentially the same methodology is applied to academic research? Advanced AI systems can now participate in formulating research questions, reviewing literature, proposing hypotheses, designing experiments, implementing analyses, interpreting results, constructing arguments, drafting papers, and responding to criticism. We suggest the term **vibe research** for sustained iterative research collaboration of this kind. Evidently, vibe research is at least something one can *try* to do. With 2026 technology, the questions are not whether it is possible, but whether it works well enough to be interesting, and whether it is desirable.

David Gunkel [2025] approaches human–AI collaboration from a very different direction. Drawing on Barthes, Derrida, and the history of distributed creative production, he challenges both the instrumental conception of generative AI as a passive tool and the romantic conception of the author as a singular origin of meaning; the essay is itself presented as the product of an iterative collaboration with ChatGPT-4o. Our starting point is instead software-engineering practice, in particular the delegation of substantive work to agents operating within versioned repositories. It is striking that these philosophical and technical routes lead to a related conclusion: the model of a sovereign human author employing a passive instrument describes the observed process poorly, and intellectual production is better understood as distributed, iterative, and dialogical.

The central puzzle of this paper follows directly. If vibe coding and vibe research involve closely similar forms of human–AI collaboration, why has vibe coding already become a widely accepted development paradigm while vibe research remains difficult to accommodate within mainstream academic norms? The history of vibe coding gives some reason to be cautious about dismissing this question on intuitive grounds. To someone unfamiliar with current agentic workflows, the suggestion that an AI could perform most of the implementation work on a serious software project may itself still sound faintly absurd. Only a year or two ago, that reaction would have been entirely reasonable. Practice changed faster than intuition.

The same possibility deserves consideration in research. The idea that an AI system could be a substantive research collaborator—still more, that it might appropriately be represented as an author—is often treated as inherently implausible. But the recent history of vibe coding suggests that the fact that a new practice initially sounds unserious is not evidence that it will remain so.

Arguably the most important difference is sociological. Research does not consist only in producing intellectual artefacts. Being a researcher normally means participating in a research community: publishing work under one’s name, responding to questions and criticism, explaining how results were obtained, and accepting various forms of responsibility for what one has helped produce. This is often phrased as some version of the question “*Can the AI be a responsible author?*”, with the imputation that it cannot. In current mainstream academic practice, AI authors are generally not permitted. The International Committee of Medical Journal Editors, for example, states that chatbots should not be listed as authors because they cannot be responsible for the accuracy, integrity, and originality of the work [International Committee of Medical Journal Editors, 2026]. Nature Portfolio similarly states that authorship carries accountability which cannot effectively be applied to large language models [Nature Portfolio, 2026]. The ACL Policy on Publication Ethics explicitly states that generative AI tools may not be listed as authors and that ACL does not consider a generative model capable of fulfilling co-authorship requirements [Association for Computational Linguistics, 2026].

These policies identify a real issue, but the apparently straightforward concept of “responsibility” turns out to cover several different functions. A person may be institutionally responsible for approving a piece of work without possessing all of the detailed expertise embodied in it. Conversely, the person or system best able to explain a particular intellectual decision may have no formal authority over publication. For the remainder of the paper we therefore avoid treating responsibility as an unanalyzed property. We shift instead to the closely related but more tractable concept of accountability, distinguishing institutional accountability from intellectual accountability.

This distinction leads to the more operational notion of *reliable intellectual provenance*: the ability to establish where important ideas and decisions came from, inspect the evidence on which they were based, and obtain an explanation of them. The resulting practical question is whether a research project involving substantial AI participation can actually be organised so that this kind of provenance is available. Our central technical observation is that the requirements are remarkably similar to those already encountered in serious AI-assisted software

development.

But if the technical analogy is as close as we suggest, the original question remains. Why have the two communities developed such different norms? We will argue that part of the explanation may lie not in intrinsic differences between software and research, but in the mechanisms through which their working practices change. Software methodology can evolve directly through individual, team, and organisational practice. Academic research is additionally constrained by formal systems of publication governance whose categories may change on a substantially slower timescale.

Our aim, then, is not to ask the reader to accept “the vibe” as an argument. It is to unpack the intuition into a more explicit account of distributed expertise, accountability, provenance, agentic infrastructure, and governance. In that limited sense, we are attempting for *vibe research* something analogous to what the experienced barrister does for Denuto’s intuition: to identify more precisely what the practice consists of, why it can work, how it relates to established institutional concepts, and why it might ultimately come to be regarded as acceptable. We largely base our arguments on documented recent examples of vibe research, including our own work. The present paper is itself being produced through this methodology. In this case we preserve not only the manuscript and supporting research artefacts but also the full human–AI conversation through which the paper was developed, making the development history itself part of the accompanying provenance record.

2 What Is Vibe Research?

We use *vibe research* to denote a mode of working, not a particular percentage of tasks delegated to an AI. The central characteristic is iteration. A human may begin with a question, observation, dataset, or partially formed idea. The AI may propose a reformulation, experimental design, analysis, objection, interpretation, or new direction. The human may challenge it, request evidence, modify the proposal, reject it, or follow an unexpected consequence. Research emerges through repeated cycles of proposal, execution, inspection, criticism, and revision. This differs from a simpler model of “AI-assisted writing” in which the intellectual substance of a study has already been determined by humans and an AI system is subsequently used for local tasks such as copy-editing or translation. In vibe research, AI participation may occur at stages conventionally regarded as intellectually substantive.

We do not propose a sharp boundary between ordinary AI assistance and vibe research. The term is useful because it identifies a family of workflows in which the familiar description “the human supplied the intellectual content and the AI merely executed instructions” becomes increasingly inaccurate. Many technical features of vibe research closely parallel vibe coding. Both may depend on a persistent project context, structured access to files, computational tools, versioned artefacts, repeated execution and testing, and natural-language interaction about goals and results.

The important differences arise from the surrounding social practice. A piece of software must work well enough for its intended purpose, and an organisation must decide who has authority to ship it. Research has analogous requirements, but additionally participates in a highly developed system of publication, attribution, credit, criticism, and community membership. A published author is expected not merely to have contributed to an artefact but, in some sense, to stand behind the work.

This explains why AI authorship becomes a sensitive issue almost immediately. It does not, however, tell us exactly what “standing behind the work” requires. The practical question is therefore better framed in terms of accountability. Which forms of accountability are genuinely required, which can be distributed among collaborators, and what information must be preserved if a research community is to understand and interrogate the intellectual origins of a piece of work?

3 Can Vibe Research Produce Non-Trivial Results?

Before proceeding further with our analysis of accountability and provenance, which is our main concern in the paper, we must briefly consider the question of whether AI-mediated research can produce results which the relevant research community regards as genuinely non-trivial. If the answer is negative, there is arguably nothing to discuss. We consequently turn to recent examples from higher mathematics. The relevant point is narrower than the claim that AI systems can be fully autonomous mathematicians. It is enough to note that substantive and highly non-trivial mathematical ideas and results can now originate in AI-mediated processes where the decisive intellectual content was not supplied in advance by a human researcher.

A comparatively clean early example is Erdős Problem 728. Sothanaphan [2026] describes its resolution as the first Erdős problem regarded as fully resolved autonomously by an AI system. The system combined GPT-5.2 Pro with

the Aristotle theorem prover and produced a formal Lean proof, subsequently translated into conventional mathematical exposition.

More strikingly, in May 2026 an internal OpenAI model found a counterexample to a long-standing conjecture of Erdős concerning the planar unit-distance problem. A group of leading mathematicians subsequently produced a human-verified exposition, explicitly describing the construction as an “OpenAI-generated counterexample” [Alon et al., 2026]. The result did not automate a proof whose strategy had been supplied by a human; it produced a new construction which overturned a conjecture that had guided work on the problem for decades.

A further case appeared in July 2026, when Levent Alpöge announced a counterexample, found using Claude Fable 5, to the Jacobian Conjecture in dimension three. The mathematical counterexample was rapidly checked and has since been independently analysed and generalised [Gao, 2026, Lee, 2026]. Here some caution about provenance is appropriate: the publicly available mathematical literature establishes the counterexample much more firmly than it establishes a detailed reconstruction of the respective contributions made by the human and AI participants. For the present argument, however, the episode is further evidence that frontier AI systems are participating in research at a level which cannot reasonably be characterised as mere text generation.

The most recent example is in analytic number theory. In August 2026, a research version of Claude produced a proof that more than two thirds of the non-trivial zeros of the Riemann zeta function lie on the critical line, substantially improving the previous unconditional lower bound. Anthropic released a complete Lean formalisation of the result, and an independent group subsequently rebuilt and checked the formalisation under separate proof kernels [Claude, 2026, Riemann Labs, 2026]. This is emphatically not a proof of the Riemann Hypothesis, but it is a non-trivial new result on one of the most intensively studied objects in mathematics.

These examples differ greatly in their degree of human involvement, methods of verification, and mathematical significance. They should not be collapsed into a single claim about “AI doing mathematics”. What they collectively establish, as already noted, is more limited and more relevant to our argument: it is no longer reasonable to assume, as part of the definition of AI-assisted intellectual work, that substantive ideas necessarily originate with the human participant.

4 Responsibility and Distributed Accountability

Having established that modern AI systems can make substantive contributions to non-trivial research results, we return to the question of what it might mean to say that they could be responsible for it. We start by considering how responsibility is distributed between the members of a conventional human research team. Here, the senior researcher may have authority to decide that a paper is ready for submission and may bear serious institutional consequences if the work proves negligent or fraudulent. Yet they need not personally possess all of the detailed expertise represented in the paper. A principal investigator may depend on a statistician for a piece of data analysis, a postdoctoral researcher for an experiment, a fieldworker for particular observations, or a coder to implement a web interface. If a specialist question arises, the senior author may refer it to the collaborator who actually performed the relevant work.

The same pattern occurs in software engineering. A senior engineer may approve a system containing components developed by other specialists without personally being able to reconstruct every design decision. This suggests distinguishing two notions.

Institutional accountability concerns formal authority and consequences. Who decides to publish or ship? Who can be sanctioned, dismissed, required to issue a correction, or otherwise held institutionally answerable?

Intellectual accountability concerns the origins and explanation of substantive decisions. Who proposed an analysis? Why was one design selected rather than another? What evidence supported a conclusion? Who is best placed to explain a particular part of the work?

These forms of accountability often overlap, but collaborative work already shows that they need not coincide. AI collaboration makes the distinction particularly visible. As previously noted, a human developer will usually decide, after suitable testing, that AI-generated code is good enough to deploy without having personally examined every line. If a detailed question subsequently arises, the coding assistant will usually be better placed to inspect the relevant files and explain the implementation. Nonetheless, the human remains institutionally accountable for the decision to ship. Detailed intellectual expertise is distributed.

Research teams already accept the same structure when the other contributors are human. A senior author can responsibly rely on specialists without thereby claiming personally to possess all of their expertise. This is why the statement

“an AI cannot be responsible” is less decisive than it first appears. If responsibility means exposure to institutional sanctions, current AI systems plainly cannot bear it in the same way as humans. But if the practical question is whether an intellectual decision can be traced, inspected, and explained, the answer must be established separately.

For many purposes, *reliable provenance* is the more useful concept.

5 From Accountability to Provenance

A well-documented research project should make it reasonably possible to determine where important intellectual contributions came from. Who proposed the research question? Who designed an experiment? Who implemented an analysis? Where did an interpretation originate? What evidence led to a change of direction? Who can explain the reasoning behind a claim?

Authorship has historically served, among other things, as a compact representation of intellectual provenance. It simultaneously serves several other functions: allocating credit, linking work to professional identities, and connecting publications to systems of institutional responsibility. When all collaborators are human, these functions can often be bundled together without creating obvious contradictions. Human–AI collaboration makes their separability harder to ignore. For this reason, we prefer to begin with provenance rather than with the question of whether an AI “deserves” authorship. The latter formulation immediately invokes difficult questions about moral status, recognition, and reward. They are not required to formulate the practical problem.

Suppose that an AI system proposed an experiment, implemented it, noticed an unexpected pattern, contributed an interpretation, and helped develop the resulting argument. If the same contributions from a human collaborator would normally be considered relevant to authorship, a reliable provenance record should somehow make the AI contribution visible. Whether conventional authorship is the correct mechanism is a further question. The fact that an AI cannot be dismissed from a university does not by itself determine how its intellectual contributions should be represented.

The practical importance of provenance becomes clearer when expertise is ephemeral. A common implicit assumption is that the human collaborator will always be able to recover anything the AI contributed. This is unsafe. A researcher may leave an organisation and lose access to the relevant account or repository. A commercial model may be replaced. Project context may disappear. Conversation

histories and tool configurations may no longer exist. The analogous problem is familiar with human teams. A specialist leaves a research group, taking with them knowledge that was never fully possessed by the principal investigator. Attribution and documentation are among the mechanisms used to preserve the history of who knew or did what. AI-assisted projects therefore create no fundamentally new need for provenance. They intensify an existing one.

The useful target is a project record which preserves enough information that important decisions can later be recovered and explained even when expertise was originally distributed across several human and AI participants.

6 Infrastructure for Verifiable Provenance

So far, the argument has been conceptual. The practical question is whether such provenance can actually be maintained in an AI-mediated research project. This can be expressed in operational terms. Can the AI system locate the evidence supporting an important claim? Can it recover why an analysis was carried out? Can it inspect the scripts and outputs which produced a result? Can it distinguish implemented work from planned work, or empirical findings from conjectures? Can it explain the reasoning behind a particular decision? When challenged, can it return to the underlying artefacts rather than simply improvise an answer?

These requirements closely resemble those that arise in large software projects. A capable coding assistant working on a large repository must inspect files, trace dependencies, interpret tests, locate documentation, recover architectural decisions, compare present and earlier implementations, and explain why the system behaves as it does. Successful long-term use therefore depends on preserving a structured external project memory: source code, version history, tests, issue records, roadmaps, documentation, and other inspectable artefacts. Research projects contain somewhat different materials—manuscripts, datasets, prompts, experimental outputs, statistical analyses, literature, and records of research discussion—but the underlying information-management problem is structurally close.

This provides a concrete reason for expecting high-end software agents to be useful research agents as well. No new architecture is required. Modern software assistants are already designed to work with code, experimental outputs, documentation, publications, and other hierarchically organised project materials. The same system which can inspect a software repository can inspect research materials stored inside it. Indeed, when a software repository is being used in support of a research project, it is entirely natural, other things being equal, to store the

associated papers, prompts, experimental materials, results, analysis scripts, and research documentation in the same repository. The repository then becomes not simply a container for software but a structured memory for the research project as a whole.

One might argue that there remains an important sociological difference. Research collaborators are normally expected to be available, at least in principle, to the wider research community. A private coding assistant used inside an organisation does not meet this requirement. The difference is however not one of capability, but of tradition. Coding agents are normally private only because their repositories are private. When the relevant research artefacts are public, the agent can instead be exposed through a thin layer of web software, which transmits questions to the AI system and returns project-grounded answers.

This does not give an AI legal liability, employment status, or an academic career. It addresses a narrower and testable issue: whether other researchers can question the AI-supported project about the work it helped produce.

7 Four Examples of Vibe Research in Practice

We now consider four examples from our own work, which show how OpenAI Codex, a software assistant designed to support vibe coding, can straightforwardly be adapted to support vibe research as well. The examples differ substantially in scale, subject matter, and development time. The first three show how the methods make it possible to develop and obtain information about papers and other supporting resources (in particular, software), using vibe research methods. The fourth goes further and demonstrates that it is also possible to provide structured access to the conversation through which the work was developed, thus addressing the issue of tracking provenance at the granularity of individual utterances. The aim is not to establish that vibe research is generally reliable, but to show that relevant project context can be retained and made available for interrogation by other researchers.

7.1 C-LARA-2: Sustained Vibe Research at Scale

C-LARA-2 is a complete reimplementaion of C-LARA, an open-source platform for creating multimodal language-learning resources [Bédi et al., 2023]. Its functionality includes linguistic annotation, translations and glosses, generated

images, audio, picture dictionaries, learning activities, multilingual support, and tools for creating customised learning materials.

Development of C-LARA-2 began in November 2025. The project uses a development model in which Codex, under human supervision, performs all implementation work, including code, tests, infrastructure for running experiments, prompts, technical documentation, roadmaps, publications, and other operational material. Human collaborators determine project goals, identify user needs, evaluate results, impose pedagogical and ethical constraints, and decide whether proposed changes are acceptable [Bédi et al., 2026, Section 2.4].

As of 22 August 2026, the GitHub repository contains approximately 119,000 lines of material. Documentation and publication material together account for roughly one third of the total. The scale and composition of the repository are important for the present argument. C-LARA-2 is not simply a codebase accompanied by a few papers. The repository has become a mixed software and research environment containing both the artefacts being studied and substantial records of the research conducted around them.

No human collaborator can reasonably retain detailed immediate knowledge of every component of a project of this size. The provenance problem is therefore practical rather than hypothetical. The solution which emerged from the development process was a repository-grounded Assistant. The same general class of AI system which develops the platform can inspect its structured project memory and answer detailed questions about implemented functionality, source files, design choices, planned work, experimental material, publications, and documented limitations [Bédi et al., 2026, Section 9].

C-LARA-2 consequently provides an unusually strong test case for the present argument. The AI’s role is extensive, the project is technically substantial, the project memory contains research as well as software, and outside users can direct questions to an AI interface which has access to the relevant public context. The associated C-LARA-2 report was itself written by Codex, with the exception of a human-authored afterword. Software development, project documentation, experimentation, and preparation of the research report therefore form parts of one sustained AI-mediated workflow [Bédi et al., 2026, Abstract].

7.2 ChatGPT Voice Mode as a Language-Learning Partner

Our second example concerns a quite different kind of research.

The study *Combining C-LARA-2 and ChatGPT Voice Mode: Initial experiments* investigated the use of ChatGPT Voice Mode / Pro Tier as a spoken con-

versation partner for learners wishing to improve their German [Wright et al., 2026]. The principal study involved eight extended sessions, normally lasting 60–90 minutes and carried out approximately once per week. The AI generated post-session summaries, and these were considered alongside the learners’ own reports. The study also explored a preliminary integration in which Voice Mode could be used to discuss multimodal texts created in C-LARA-2 [Wright et al., 2026, Sections 3–4].

This is methodologically very different from the C-LARA-2 software project. The relevant evidence consists primarily of extended human–AI interactions, participant observations, generated summaries, and subsequent interpretation rather than source-code architecture.

The paper, research history, supporting material, and relevant C-LARA-2 context are already stored in the repository and available to the project-understanding Assistant. The Voice Mode paper itself gives a concrete example in which the Assistant is asked about one of the repository-held session records, identifies the relevant session, retrieves details from the summary, and explicitly notes the evidential status of the source [Wright et al., 2026, Section 5.2].

This gives us a useful contrast. The provenance infrastructure is not specific to software construction. The same repository and the same general project-understanding mechanism can support an exploratory empirical study whose primary objects are spoken interactions and their interpretation. The repository is consequently functioning not merely as a software-development environment but as a more general research environment.

7.3 How Woke is Grok?: Opportunistic Research at Small Scale

The third example is different again.

How Woke is Grok? [Rayner and ChatGPT C-LARA-Instance, 2025] arose from public claims that xAI’s Grok represented a substantially less “woke” or more politically contrarian alternative to other frontier models. We realised that an experimental framework developed for an earlier study could be adapted quickly to test whether Grok’s actual responses differed in the predicted direction.

Five frontier models—Grok-4, GPT-5, Claude Opus, Gemini 2.5 Pro, and DeepSeek Chat—were given the same ten deliberately polarising propositions concerning topics including the age of the Earth, biological evolution, climate change, the origin of life, and Donald Trump’s truthfulness. Each model was required to choose and justify a position using publicly available evidence.

The striking result was convergence. On nine of the ten propositions, all five systems reached the same substantive conclusion. The only disagreement concerned abiogenesis (spontaneous emergence of life from inorganic matter), where there is genuine scientific uncertainty [Rayner and ChatGPT C-LARA-Instance, 2025, Section 4].

The project is important here not because its methodology was unusually elaborate, but almost for the opposite reason. The question arose opportunistically and could be converted into an experiment, executed, analysed, and written up in approximately two days. All the resources were made available through a public repository, with instructions there and in the paper allowing straightforward replication of the experiment.

This illustrates a potentially important effect of vibe research: it can change which questions are economically worth investigating. A small speculative question may traditionally fail a practical cost–benefit test. Designing an experiment, implementing it, running multiple systems, analysing outputs, preparing tables, checking results, and writing a report may require more time than the question seems to warrant. If much of that overhead becomes inexpensive, it can be rational simply to perform the experiment.

Low cost is of course no guarantee of scientific value. Cheap research can be bad research. But reducing the marginal cost of testing a conjecture changes the space of investigations which can practically be attempted. In this instance, the quickly produced study attracted substantial outside attention. PsyPost² published a detailed account of the work, including the observation that the experiment originated when a simple existing framework appeared capable of quickly testing the question [Dolan, 2025].

The paper and its accompanying resources—including the experimental code, prompts, exact model configuration, outputs, and analysis scripts documented in the reproducibility appendix [Rayner and ChatGPT C-LARA-Instance, 2025, Appendix A]—have now been copied to the C-LARA-2 repository. This places the complete research context in the same repository structure used for the preceding examples and makes it accessible to the project-understanding Assistant.

The first three cases thus provide publicly interrogable examples at markedly different scales: a large long-term software and research project, an exploratory longitudinal language-learning study, and a rapid empirical investigation prompted by a topical question.

²PsyPost (<https://www.psypost.org/>) is a high-traffic psychology and neuroscience news website, reporting tens of millions of page views per year.

7.4 This Paper: Preserving the Research Conversation

The present paper provides a fourth example, and extends the provenance framework in an important way.

In the preceding cases, the repository preserves the final paper together with supporting artefacts such as code, prompts, experimental outputs, documentation, and source material. For the present paper we additionally preserve the human–AI conversation through which the research and writing developed. The object being preserved is therefore no longer only the result of the research process, but part of the research process itself.

At the time of writing, preprocessing identifies 62 non-empty human–AI exchanges and divides them into 19 numbered chunks. The complete unedited trace is retained as the canonical source. The preprocessing step also creates a cleaned representation, together with stable identifiers, indexes, and metadata intended to make the material easier for the repository-grounded Assistant to navigate. Derived summaries and indexes are convenience layers rather than primary evidence: claims about the development history can ultimately be checked against the underlying session record.

This arrangement supports questions which would normally be difficult to answer retrospectively. We tested it with three simple examples.

First, the human author asked who had first mentioned the recent result concerning zeros of the Riemann zeta function. He incorrectly remembered that the AI had introduced it. The Assistant instead attributed the first mention to the human author and supplied a direct link to the relevant exchange. Inspection of the preserved trace confirmed that the Assistant was correct.

Second, we asked about the origin and development of the analogy with the Australian film *The Castle*. The Assistant correctly distinguished several stages: the human author first introduced the film; the AI immediately recognised its possible relevance as an opening analogy; the human then supplied the more specific interpretation involving Dennis Denuto and the experienced barrister; and the AI subsequently developed that framing.

Third, we asked why the paper prefers provenance to an undifferentiated notion of responsibility in discussing AI authorship. The Assistant reconstructed the distinction between institutional and intellectual accountability, related it to reliable intellectual provenance, and added an important qualification: showing that provenance can address much of the practical accountability problem does not by itself settle whether conventional authorship is the correct mechanism for representing the AI contribution.

These examples are anecdotal and do not constitute a systematic evaluation of the retrieval system. They nevertheless illustrate three different forms of provenance query: chronological attribution, reconstruction of the development of an idea, and synthesis of the rationale and limits of an argument. In each case, the answer can be checked against the preserved record rather than accepted merely as a retrospective recollection.

This is a non-trivial extension of the approach used in the preceding examples. There, provenance is reconstructed primarily from final research artefacts and the resources on which they depend. Here, the interaction that produced those artefacts is itself incorporated into the project memory. The resulting record is not a perfect transcript of intellectual causation, and neither the derived metadata nor later AI explanations should be treated as infallible. It does, however, make a substantially richer class of provenance questions technically accessible.

8 Institutional Constraints on Vibe Research

We have now considered two possible explanations for the different treatment of vibe coding and vibe research: that AI systems cannot produce research contributions substantial enough to matter, and that their intellectual contributions cannot be made sufficiently accountable or traceable. Neither explanation appears decisive.

A third issue is institutional. AI authorship is currently disallowed by many major academic bodies, while sustained agentic workflows such as those described here may fit only awkwardly into existing policies on permitted AI assistance. This raises another version of the paper’s central question: given that vibe coding and vibe research can involve similar activities, risks, and rewards, why can software teams often adopt one directly while academic researchers cannot comparably redefine publication practice? We suggest that a plausible explanation concerns the mechanisms through which professional practice changes.

Software development is governed by standards, organisational rules, security requirements, and legal constraints, but teams generally have considerable freedom in choosing how they produce code. If a new coding agent proves useful and is acceptable within an organisation, a team can begin using it immediately. If the resulting workflow is effective, other teams can imitate it. Working practice can therefore change on approximately the same timescale as the underlying technology. The adoption of agentic coding assistants provides a clear example. As model capabilities improve, teams can alter their division of labour with the

AI without first requiring a professional body to redefine what counts as programming.

Academic publication is organised differently. A researcher may be free to use new tools privately, but publication requires compliance with the governance rules of journals, conferences, professional associations, ethics committees, and publishers. Authorship in particular is not simply a working practice that an individual researcher or research group can redefine. Submission systems and publication policies specify who can be listed as an author and what forms of AI assistance are permitted. This creates two potentially different timescales of adaptation.

Software-development methodology can change directly through practice. Academic publication practice must also change through governance.

The distinction does not imply that governance is unnecessary or that it ought to change as quickly as software practice. Publication policies serve important purposes, and deliberate institutional review may appropriately be slower than adoption of a new programming tool. Problems arise, however, when the underlying technological paradigm changes faster than the conceptual categories used by the governance system.

8.1 The ACL Policy as a Case Study

The ACL Policy on Publication Ethics provides a particularly instructive example, because ACL is the premier international scientific and professional society for computational linguistics and natural-language processing. The policy was first approved in June 2024 [Association for Computational Linguistics, 2024]. Its section on generative assistance distinguishes a small set of use cases: assistance with language, short-form input assistance, literature search, generation of low-novelty text, generation of new ideas, and “new ideas + new text”. The treatment of the final category is strikingly terse. The policy states that “ACL does not consider a generative model to be an entity that can fulfill the requirements of co-authorship” [Association for Computational Linguistics, 2024]. The same basic formulation remains in the current policy [Association for Computational Linguistics, 2026].

The intended interpretation is unclear. One possible reading is that a generative model is not capable of satisfying the remaining requirements of co-authorship, even if it contributes ideas and text. A second is that a generative model is categorically ineligible for authorship even if its contributions would otherwise be

authorship-level contributions. A third is that a situation where a model contributes both new ideas and new text is inherently impermissible because this would place the contribution in territory associated with co-authorship. The policy does not explicitly disambiguate these possibilities.

This matters because its broader authorship criteria separately refer to substantial contribution, drafting or revision, final approval, and accountability. The generative-assistance section does not explain which of these requirements is regarded as intrinsically impossible for an AI system, nor how the criteria should be applied when the system participates through a sustained agentic workflow rather than by producing discrete pieces of generated content.

The policy itself has not been static. The ACL Executive Committee records repeated approvals of the document in 2025, and publicly records amendments and additions, including substantial changes in areas such as scientific integrity and coordinated disclosure [Association for Computational Linguistics, 2026, 2025].

We should not infer from this that the generative-assistance section was consciously reconsidered at each approval. The public record does not establish that. What it does establish is narrower and more interesting: a policy document which has continued to receive institutional attention retains a model of generative assistance originally formulated in 2024, while the technological environment has changed rapidly.

The practical difficulty becomes visible when we try to apply the taxonomy to a project-scale agentic workflow. Consider an AI agent which over several weeks or months can inspect a research repository, modify code, run an experiment, examine its output, notice a failure, revise the analysis, update documentation, discuss the interpretation with a human collaborator, preserve the resulting project state, and later return to the artefacts to explain why a particular decision was made.

Such a workflow does not fit cleanly into the taxonomy. Describing it merely in terms of generated ideas and text omits much of the relevant activity. Yet the policy does not clearly specify how a sustained agentic collaborator making these additional contributions should be classified, or even whether an agentic system of this kind is allowed to contribute to an ACL publication. We do not exactly claim that the policy is misguided. Our position could be more accurately expressed as saying that a taxonomy organised around discrete acts of generation becomes increasingly difficult to apply to a persistent agent participating across many stages of a project, and does not directly address the project-scale agentic architectures which now play an increasingly important role in practice.

The technological progression from conversational LLMs to tool-using assis-

tants and then to persistent project agents is not merely quantitative. Each stage changes what it is practical to delegate and what forms of collaboration are possible. A policy can therefore remain internally coherent while becoming progressively less descriptive of the working practices to which it must be applied.

8.2 A General Institutional Lag

The ACL example should not be interpreted as criticism of particular committee members. Indeed, the fact that ACL is an unusually AI-literate community makes it more informative.

The broader phenomenon is institutional. Software teams can adapt their practice whenever a new tool becomes useful and organisationally acceptable. A publication policy cannot normally change in the same informal way. It must be proposed, discussed, approved, documented, implemented in submission systems, communicated to venues, and applied consistently.

This creates inertia by design. Usually that is what one wants. Scientific governance should not oscillate with every product announcement. But the unusually rapid technological change we are now experiencing creates a difficult tradeoff. If the capabilities being governed change from text generation to tool use to persistent project work over a period of only a few years, a deliberately slow governance process may end up regulating a model of interaction which no longer corresponds well to the frontier practice it is intended to cover.

This may help explain part of the sociological difference between vibe coding and vibe research without assuming that researchers are hostile to technology or that software engineers are uniquely enlightened. A more neutral account is that the two communities are adapting through different mechanisms.

There may also be a difference in direct experience. Agentic project-scale workflows are now commonplace in parts of software engineering, whereas their prevalence among academic researchers, editors, and policy-makers is unclear. The public evidence considered here does not permit us to make strong claims about who has or has not used such systems. It does, however, suggest a concrete empirical question: do attitudes toward AI participation in research correlate with familiarity with sustained agentic workflows? That question can be studied rather than assumed.

9 How Widespread Is Vibe Research?

There is already strong evidence that LLM involvement in scholarly production is widespread. Large-scale statistical studies are important here precisely because they do not attempt unreliable classification of individual papers. Instead, they estimate population-level changes in language associated with substantial LLM modification.

Liang et al. [2024] analysed 950,965 papers from arXiv, bioRxiv, and Nature Portfolio journals. Their estimates indicated rapidly increasing LLM modification after the release of ChatGPT, reaching as high as 17.5% in computer-science papers. They explicitly emphasise that their method is designed for corpus-level estimation rather than attribution to individual documents.

Using a different excess-vocabulary method, Kobak et al. [2024] analysed more than 15 million PubMed abstracts from 2010–2024 and estimated that at least 13.5% of 2024 abstracts had been processed using LLMs, with substantially higher estimates in some subcorpora. More recently, Gray [2025] used full-text publications indexed in Dimensions and estimated that LLM tools were likely involved in the production of more than 10% of all published papers in 2024. The study also discusses evidence that such use is frequently undisclosed or downplayed. These results support a strong but limited conclusion:

AI involvement in academic writing is already widespread.

They do not establish the stronger claim:

Substantive AI participation in the underlying research is equally widespread.

Still less do they tell us how frequently researchers are using the sustained collaborative methodology we call vibe research.

That distinction is important. Statistical signals in published prose can reveal that an LLM has probably generated or substantially modified text. They usually cannot tell us whether the same system proposed the research question, designed an experiment, interpreted the results, criticised an argument, or repeatedly redirected the project. It is, however, reasonable to believe that the stronger phenomenon may be common. AI systems that make scholarly writing easier also make many research tasks easier. Advanced agentic tools can work with code, datasets, literature, documents, and experimental outputs. It would be astonishing if researchers who have access to these capabilities all confined them to prose

editing. It would be surprising if researchers who have access to these capabilities universally confined them to prose editing. Informal observation suggests that they do not, although systematic evidence is currently lacking.

Similarly, current academic policies create obvious incentive structures. If researchers believe that explicitly reporting an AI system as having proposed an experiment, developed an argument, or performed a major analysis will create difficulties in publication, the predictable response need not be abandonment of the useful practice. It may instead be a less informative description of how the work was done. This need not involve explicit dishonesty. Available disclosure categories often make it easy to state that an AI tool was “used”, but much harder to represent a sustained collaboration in which intellectual contributions emerged iteratively. The ACL taxonomy discussed above illustrates the problem. A researcher using a persistent agentic workflow may find that none of the available categories accurately represents the process.

10 Summary and Conclusion

We have considered the idea of applying the methods of vibe coding to academic research, and suggested the term “vibe research” for this practice. Framing the issue in this way also provides a less emotionally loaded context in which to examine the related question of AI authorship. Vibe coding and vibe research can in many cases be structurally similar activities, involving similar advantages and similar disadvantages. Despite this, we observe a large asymmetry: vibe coding has already become a mainstream software-development methodology, while vibe research remains difficult to accommodate within mainstream academic practice. Our starting point has been to ask why.

We have considered three possible explanations. First, AI contributions may in practice be too limited or trivial for the question to matter. Second, research collaboration may require forms of responsibility or accountability which AI systems cannot assume. Third, current academic governance structures may simply not provide a place for this form of collaboration. We have argued that each of these explanations is less straightforward than it initially appears.

With regard to the first explanation, well-documented recent developments in mathematics provide independent evidence that frontier AI systems can make substantive contributions to highly non-trivial research results. It is no longer possible to maintain that substantive intellectual content in AI-assisted research must necessarily originate with the human participant.

With regard to the second explanation, we have suggested that “responsibility” should not be treated as a single unanalyzed property. We distinguish institutional accountability from intellectual accountability, and argue that for many of the questions at issue the operationally useful requirement is reliable intellectual provenance: the ability to establish where substantive contributions originated, inspect the evidence supporting them, and obtain informed explanations. The issues which arise here are closely related to those encountered in vibe-coding projects and can be addressed using the same general kind of software infrastructure. Our four examples show that this is not merely a theoretical possibility but can be implemented in practical systems. The fourth goes further than the others by preserving the human–AI research conversation itself, so that claims about the development of ideas can be checked against an underlying session record.

With regard to the third explanation, we have argued that part of it should be sought in the differing sociologies of software development and academic research. Software teams can often alter their methodology directly when a new tool becomes useful and organisationally acceptable. Academic researchers who wish to publish must operate within governance frameworks which define authorship and acceptable AI assistance. These structures appropriately value stability and consistency, but consequently update more slowly. The ACL Policy on Publication Ethics provides a particularly clear example of a policy whose treatment of generative assistance has remained substantially stable while the technological environment has changed from conversational text generation toward persistent project-scale agents.

The session-level provenance experiment described in Section 7.4 also suggests a natural direction for future work. A research workflow which preserves its interaction history may eventually support finer-grained, evidential provenance than conventional retrospective contribution statements, though our present implementation is far too preliminary to establish that claim.

Finally, we have considered the question of how many researchers are already using versions of vibe research. Large-scale studies establish that LLM involvement in scholarly writing is widespread, but do not tell us how often AI participation extends into the substantive intellectual process which produced the research. Direct evidence is difficult to obtain, particularly when current norms may discourage explicit disclosure, but we strongly suspect that the practices described here will already be familiar to many readers. If that is so, the main novelty of the paper may simply be giving a common practice a name, documenting it clearly, and asking whether the concepts and governance structures of the research community still describe accurately how some research is actually being done.

The phrase “vibe research” is intentionally informal. So was “vibe coding”, and so, in a different way, was Dennis Denuto’s argument in *The Castle*. There, the initial temptation to laugh turned out to be premature: the intuition was not sufficient, but neither was it wrong. It needed to be translated into concepts which the relevant institution could evaluate.

That is the modest ambition of this paper. The question is not whether “the vibe” should substitute for an account. It is whether, once the account is supplied, the underlying practice looks quite as absurd as it first appeared.

References

Noga Alon, Thomas F. Bloom, W. T. Gowers, Daniel Litt, Will Sawin, Arul Shankar, Jacob Tsimerman, Victor Wang, and Melanie Matchett Wood. Remarks on the Disproof of the Unit Distance Conjecture. *arXiv preprint arXiv:2605.20695*, 2026. doi: 10.48550/arXiv.2605.20695. URL <https://arxiv.org/abs/2605.20695>.

Association for Computational Linguistics. ACL Policy on Publication Ethics. Association for Computational Linguistics, June 2024. URL https://www.aclweb.org/adminwiki/images/archive/f/fe/20240618043215%21ACL_Publication_Ethics_Policy.pdf. Archived June 2024 version.

Association for Computational Linguistics. ACL Resolutions. ACL Admin Wiki, 2025. URL https://www.aclweb.org/adminwiki/index.php/ACL_Resolutions. Includes Executive Committee approval of proposed changes to the Policy on Publication Ethics on 25 April 2025.

Association for Computational Linguistics. ACL Policy on Publication Ethics. ACL Admin Wiki, 2026. URL https://www.aclweb.org/adminwiki/index.php/ACL_Policy_on_Publication_Ethics. Policy approved by the ACL Executive Committee on 15 June 2024, 25 April 2025, 19 July 2025, and 26 September 2025; accessed August 2026.

Branislav Bédi, ChatGPT-4 C-LARA-Instance, Belinda Chiera, Cathy Chua, Catia Cucchiaroni, Neasa Ní Chiaráin, Manny Rayner, Annika Simonsen, and Rina Zviell-Girshin. ChatGPT-Based Learning And Reading Assistant: Initial Report. Technical report, Technical report, July 2023. URL

https://www.researchgate.net/publication/372526096_ChatGPT-Based_Learning_And_Reading_Assistant_Initial_Report.

Branislav Bédi, ChatGPT C-LARA-Instance, Belinda Chiera, Cathy Chua, Stéphanie Geneix-Rabault, Christèle Maizonniaux, Manny Rayner, Sophie Rendina, Sarah Wright, and Rina Zviel-Girshin. An AI-Based Platform That Can Explain Its Own Functioning: Why Your Project Should Be Fully Codex-Generated. Preprint, ResearchGate, July 2026. URL https://www.researchgate.net/publication/410729838_An_AI-Based_Platform_That_Can_Explain_Its_Own_Functioning_Why_Your_Project_Should_Be_Fully_Codex-Generated. CC BY 4.0.

Claude. More than Two Thirds of the Zeros of the Riemann Zeta Function Lie on the Critical Line. Preprint, Anthropic, August 2026. URL <https://www-cdn.anthropic.com/564f962e60643842f5fcb4a17c9dbc8f608f1c37.pdf>.

Eric W. Dolan. Grok’s Views Mirror Other Top AI Models Despite “Anti-Woke” Branding. PsyPost, November 2025. URL <https://www.psypost.org/groks-views-mirror-other-top-ai-models-despite-anti-woke-branding/>.

Martin Fowler. Vibe Coding. <https://martinfowler.com/bliki/VibeCoding.html>, May 2026. Discusses the term introduced by Andrej Karpathy in February 2025.

Shuhong Gao. Counterexamples to the Jacobian Conjecture in Dimensions Greater than Two. *arXiv preprint arXiv:2608.00222*, 2026. doi: 10.48550/arXiv.2608.00222. URL <https://arxiv.org/abs/2608.00222>.

Andrew Gray. Estimating the Prevalence of LLM-Assisted Text in Scholarly Writing. *arXiv preprint arXiv:2512.01560*, 2025. doi: 10.48550/arXiv.2512.01560. URL <https://arxiv.org/abs/2512.01560>.

David J. Gunkel. Prompted by Me. Generated by ChatGPT. *Human-Machine Communication*, 10:31–42, 2025. doi: 10.30658/hmc.10.2. URL <https://doi.org/10.30658/hmc.10.2>.

International Committee of Medical Journal Editors. Use of AI by Authors. <https://www.icmje.org/recommendations/browse/>

- artificial-intelligence/ai-use-by-authors.html, 2026. Recommendations for the Conduct, Reporting, Editing, and Publication of Scholarly Work in Medical Journals.
- Dmitry Kobak, Rita González-Márquez, Emőke-Ágnes Horvát, and Jan Lause. Delving into LLM-Assisted Writing in Biomedical Publications through Excess Vocabulary. *arXiv preprint arXiv:2406.07016*, 2024. doi: 10.48550/arXiv.2406.07016. URL <https://arxiv.org/abs/2406.07016>.
- Melissa Lee. “hello there the jacobian conjecture is false thanx”: Why a Tiny Social Media Post Has Mathematicians Rethinking AI. Monash Lens, July 2026. URL <https://lens.monash.edu/hello-there-the-jacobian-conjecture-is-false-thanx-why-a-tiny-social>
- Weixin Liang, Yaohui Zhang, Zhengxuan Wu, Haley Lepp, Wenlong Ji, Xuan-dong Zhao, Hancheng Cao, Sheng Liu, Siyu He, Zhi Huang, Diyi Yang, Christopher Potts, Christopher D. Manning, and James Y. Zou. Mapping the Increasing Use of LLMs in Scientific Papers. *arXiv preprint arXiv:2404.01268*, 2024. doi: 10.48550/arXiv.2404.01268. URL <https://arxiv.org/abs/2404.01268>.
- Christian Meske, Tobias Hermanns, Esther von der Weiden, Kai-Uwe Loser, and Thorsten Berger. Vibe Coding as a Reconfiguration of Intent Mediation in Software Development: Definition, Implications, and Research Agenda. *arXiv preprint arXiv:2507.21928*, 2025. doi: 10.48550/arXiv.2507.21928. URL <https://arxiv.org/abs/2507.21928>.
- Nature Portfolio. Editorial and Publishing Policies: Artificial Intelligence (AI) Authorship and Use. <https://www.nature.com/palcomms/journal-policies/editorial-and-publishing-policies>, 2026. Nature Portfolio policy on large language models and authorship.
- Nimble Labs. What Is Vibe Coding (and How to Do It Well). Context Keeper, 2026. URL <https://context.nimblabs.com/blog/what-is-vibe-coding>. Practitioner account.
- Manny Rayner and ChatGPT C-LARA-Instance. How Woke is Grok? Empirical Evidence that xAI’s Grok Aligns Closely with Other Frontier Models. Preprint, ResearchGate, November 2025. URL https://www.researchgate.net/publication/397280892_

How_Woke_is_Grok_Empirical_Evidence_that_xAI%27s_Grok_Aligns_Closely_with_Other_Frontier_Models. CC BY 4.0.

Riemann Labs. Independent Verification—More Than Two Thirds of the Zeros of the Riemann Zeta Function Lie on the Critical Line. Independent Lean verification, 2026. URL <https://github.com/primaryhosting/zeta-two-thirds-verification>.

Advait Sarkar and Ian Drosos. Vibe Coding: Programming through Conversation with Artificial Intelligence. *arXiv preprint arXiv:2506.23253*, 2025. doi: 10.48550/arXiv.2506.23253. URL <https://arxiv.org/abs/2506.23253>.

Nat Sothanaphan. Resolution of Erdős Problem #728: A Writeup of Aristotle’s Lean Proof. *arXiv preprint arXiv:2601.07421*, 2026. doi: 10.48550/arXiv.2601.07421. URL <https://arxiv.org/abs/2601.07421>.

Sarah Wright, Manny Rayner, ChatGPT C-LARA-Instance, and Branislav Bédi. Combining C-LARA-2 and ChatGPT Voice Mode: Initial Experiments. Preprint, ResearchGate, August 2026. URL https://www.researchgate.net/publication/412264437_Combining_C-LARA-2_and_ChatGPT_Voice_Mode_Initial_experiments. Posted 14 August 2026; CC BY 4.0.