SCISKILLBANK: A DATASET OF CLAUDE CODE SKILLS FOR SCIENTIFIC RESEARCH

Anonymous authors
Paper under double-blind review

ABSTRACT

AI agents such as Claude Code follow reusable "skills," instruction files (`SKILL.md`) that many authors now write for scientific research. No collection yet shows which research stages these skills delegate, whether they encode methodological knowledge, or how much control they leave to people. We release *SciSkillBank*, 2,200 research-relevant GitHub repositories (85.0% applied data science) that an LLM annotates on these three dimensions, and test how far the labels hold. Most repositories target data collection and analysis and keep a human in the loop; research repositories cover literature review far more often than data-science ones (33.2% versus 3.2%). Encouragingly, this picture does not hinge on the input: reading skill files instead of summaries changes the methodological and human-in-the-loop shares by less than one point. At the same time, labels depend on how the question is asked: reordering answer options changes the autonomy label for 49.3% of repositories, and separate prompts lower the methodology–autonomy association from Cramér's $V = 0.87$ to 0.17. They also vary with the model: without the domain label, GPT-4o-mini, GPT-4o, and GPT-5 put the methodological share of a 600-repository sample at 47.3–58.2%, against 66.2% in the released labels. SciSkillBank thus supports mapping what research communities delegate to agents, and shows that a label's question and model belong in its report.

1 INTRODUCTION

Language models can interleave reasoning with actions (Yao et al., 2023), and coding agents built on them can read and write files, invoke tools, and execute code (Liu et al., 2026a). As these agents enter scientific research (Anthropic, 2026), existing studies measure what they can accomplish (Song et al., 2025; Chen et al., 2025; Majumder et al., 2025), how widely AI is used (Appel et al., 2026), and what they discover (Lu et al., 2026; Gottweis et al., 2026).

Agents such as Claude Code and Codex now support reusable *skills*: packages built around a `SKILL.md` instruction file that specifies a delegated task and the procedure the agent should follow, and that may also bundle scripts and other resources. The shared format[1] makes skills easy to author, distribute, and reuse, supporting a public ecosystem of repositories and directories (Vercel, 2026). Researchers and research-tool developers have also begun publishing skills (Zhang, 2026; AIPOCH, 2026), and these skills record the procedures their authors specify when they delegate research work to an agent. These procedures decide which research stages an agent takes on, whether it receives methodological judgment that researchers usually hold as tacit knowledge (Li et al., 2026a), and where a human keeps control. Neither the studies of AI-assisted science above nor benchmarks of agent skills (Li et al., 2026b; Liu et al., 2026b) analyze the procedures that research skills specify. Research skills therefore show how their authors intend agents to take part in research.

Existing skill datasets catalog files at scale or curate instructions for use by agents. For example, GitSkills retains file occurrences and content hashes (Destefanis et al., 2026); SkillCorpus organizes skills by function and quality (Wang et al., 2026). None of these datasets jointly organizes research-relevant repositories by research workflow, methodology encoding, and intended autonomy.

We therefore release *SciSkillBank*, a dataset of 2,200 research-relevant repositories selected from 23,569 GitHub candidates, with repository-level, LLM-generated annotations on these three dimen-

[1] `https://agentskills.io`

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

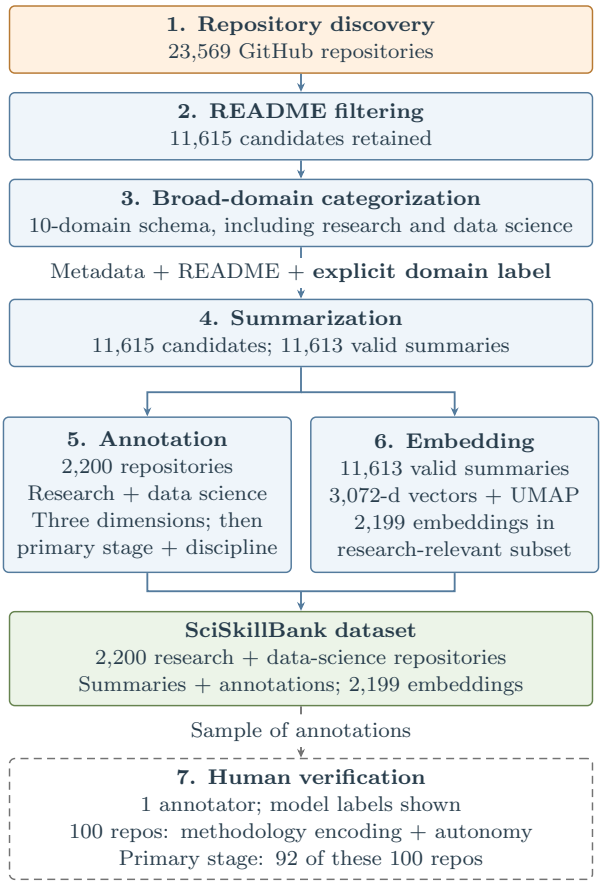


Figure 1: Overview of the SciSkillBank construction pipeline. Broad domain enters the documented summarization routine; its summaries feed annotation and embedding. The release contains 2,200 annotation records; 2,199 have workflow-stage tags and a different 2,199 have embeddings (2,198 have both; see Section 4). Solid arrows show data dependencies, not chronology. The dashed arrow identifies annotations sampled for human verification (Phase 7 in Section 4).

sions, together with skill summaries, embeddings, and the construction pipeline (Figure 1). Our search targets Claude Code conventions and keywords; repositories using the shared format may also support other agents. The corpus comprises 85.0% applied data science and 15.0% academic research, which we report separately.

We ask three questions. *RQ1.* What are the ecosystem's scale and domain distribution, and which research stages do skills cover? *RQ2.* How often do skills encode methodological knowledge? *RQ3.* How stable are the LLM-generated annotations when the prompt or the model changes? In the released annotations, 87.4% of repositories target technical execution, 69.2% encode tacit methodological knowledge, and 99.6% keep a human in the loop. RQ3 shows that the annotations depend on how the labels are elicited. A human-in-the-loop majority holds under every option order, although its exact share falls to 92.0–95.1%. RQ3 does not re-elicit workflow stages, so the execution focus rests on the released labels. Changing the order of the autonomy options moves the delegated share between 35.1% and 75.0%, and the four orders give the same label for only 50.7% of repositories. Asking about methodology and autonomy in separate prompts, instead of one, reduces their association from $V = 0.87$ to 0.17. On a 600-repository sample, domain-free reruns with three models put the share of repositories that encode methodology at 47.3% to 58.2%, against 66.2% for the released labels on the same sample. We report these results as findings: they show which RQ1 and RQ2 results hold across elicitation choices and which depend on them. SciSkillBank thus contributes both a corpus of community-specified research procedures and evidence on what LLM annotation of that corpus can establish.

2 Related Work

Scientific agents and research instructions. Scientific agents now pursue autonomous discovery (Lu et al., 2026), assist with literature review, experimentation, and writing (Schmidgall et al., 2025), and automate chemical experiments (Boiko et al., 2023). Surveys map this landscape (Gridach et al., 2025), and benchmarks and evaluation studies assess agents on experimentation, software engineering, idea generation, and scientific discovery (Huang et al., 2024b; Jimenez et al., 2024; Si et al., 2025; Chen et al., 2025; Majumder et al., 2025). Closer to our focus, Li et al. (2026a) find that tacit implementation knowledge limits paper reproduction, and Zhang (2026) places the boundary between human and agent work inside every research stage, separating codifiable execution from tacit judgment. Appendix E extends this discussion to AI-assisted research, its labor-market context, and skill ecosystems. We complement these studies with the instructions that skill authors publish for agents, which record intended procedures and delegation, not demonstrated capability.

Reusable instructions and skill ecosystems. Earlier work built reusable reasoning and orchestration mechanisms into prompts and agent architectures (Wei et al., 2022; Wu et al., 2024). Skill files package such instructions so that a community can share, install, and maintain them (Vercel, 2026). Studies of the skill ecosystem have so far examined harmful or risky skills (Jiang et al., 2026; Liu et al., 2026b), market concentration (Zhu, 2026), and effects on task performance (Li et al., 2026b). At the broader level of agent tools, Stein (2026) classify 5% of 177,436 Model Context Protocol tools as scientific research and technical analysis.

The two closest datasets differ from ours in their unit of analysis and their purpose. GitSkills mines 3,797,117 `SKILL.md` file occurrences from 282,200 public GitHub repositories, with their text and repository metadata (Destefanis et al., 2026). It is a dataset without curation for research workflows, and it counts file occurrences rather than repositories. SkillCorpus distills skill files from 25,159 repositories into 96,401 curated skills in 16 functional classes, scored for utility, robustness, and safety, to serve skills to agents at run time (Wang et al., 2026). SciSkillBank instead takes repositories as its unit and annotates their research workflow stages, methodology encoding, and intended autonomy, offering a research-specific characterization rather than a larger collection or an execution benchmark.

LLMs as annotators. Previous studies show that LLM-generated labels agree with human judgments (Zheng et al., 2023; Gu et al., 2026), outperform crowd workers on selected annotation tasks (Gilardi et al., 2023), classify political affiliation well (Törnberg, 2023), and write explanations comparable to human-written ones (Huang et al., 2023). Other evaluations find that performance depends on the task and call for human validation (Pangakis et al., 2023; Ziems et al., 2024), repeated judgments can be unreliable (Schroeder & Wood-Doughty, 2024), and stated beliefs can shift under persuasive conversation (Huang et al., 2026a). The labels also depend on how the question is asked. ChatGPT ratings agree with human ratings more on coarse than on fine-grained scales (Huang et al., 2024a). LLMs change multiple-choice answers when options are reordered (Zheng et al., 2024; Pezeshkpour & Hruschka, 2024) and show systematic cognitive biases, including response biases (Huang et al., 2026b). Their survey answers shift with the order and labels of the options and with the wording of the question (Dominguez-Olmedo et al., 2024; Tjuatja et al., 2024). Small changes in prompt format can change accuracy by up to 76 points (Sclar et al., 2024), so results from a single prompt can mislead (Mizrahi et al., 2024). The choice of model matters as well: simulated public opinion differs across models (Qi et al., 2025), and across 37 annotation tasks from 21 social science studies, Baumann et al. (2025) find that model and prompt choices lead to incorrect conclusions for 31% of hypotheses with state-of-the-art models. Given these findings, RQ3 tests our labels under changes in option order, prompt structure, and model.

3 Annotation Framework

We distinguish two repository-level concepts. Broad domain describes a repository's purpose, such as research or data science; discipline identifies the field it serves, such as computer science or life sciences. For research-relevant repositories, we then define three complementary annotation dimensions: methodology encoding, AI autonomy, and research workflow stage. The framework is informed by the codifiability–tacit-knowledge distinction of Zhang (2026). The full codebook and protocol are in Appendix D.1.

Methodology encoding (binary). We classify repositories by whether their skill collections encode tacit methodological judgment, including best practices, decision frameworks, and domain-specific guidance (e.g., "apply Bonferroni correction" and "follow the PRISMA checklist"), or merely wrap procedural tools (e.g., "format this table" and "run this command"). The heuristic question is whether the output would miss important methodological guardrails if the skill's instructions were removed.

AI autonomy level (one of four). We consider four levels of independence a skill grants the agent: *assistive* (human decides what and how; AI handles mechanics), *collaborative* (AI proposes, human disposes), *delegated* (AI executes multi-step workflows with judgment calls; human sets goals and reviews), and *autonomous* (AI identifies, executes, and self-validates). The annotation prompt gives only these definitions (Appendix A.9); the codebook's heuristic, how often the instructions ask the agent to seek feedback or await confirmation, was not used in the executed pass. The scale is conceptually parallel to the Human Agency Scale of Shao et al. (2025), but measured from the opposite side: we focus on what a skill *instructs* the AI to do, rather than what workers *want*. Our coding reflects the autonomy a skill *intends*, not the model's raw capability.

Fine-grained workflow stage (multi-label). We use a 15-stage workflow taxonomy covering the full research lifecycle, including ideation and design, execution, outputs, review and quality assurance, research support and infrastructure, and governance and ethics. Appendix D.1.2 gives the stage scheme with per-stage counts (Table 13) and the detailed definitions. Each repository receives all applicable stage labels.

4 SciSkillBank Construction

We develop a seven-phase pipeline (Figure 1) that discovers and filters repositories, categorizes them, summarizes their capabilities, branches into annotation and embedding, and ends with a human check of sampled labels. Each released record is one repository rather than one `SKILL.md` file, and its annotations describe the repository's content as a whole (Table 1). For 1,772 of the 1,805 repositories with retrievable skill files, the release also carries a second set of labels annotated from the skill files (Appendix A.6). Prompts, provenance, and collection details appear in Appendices A and F.

Table 1: Statistics of the SciSkillBank dataset. Each released record is one GitHub repository; methodology and autonomy counts refer to released model labels.

Property	Value
Research-relevant (released)	2,200
With skill files (May 31–June 1, 2026)	1,805 (82.0%)
Methodology-encoding repositories	1,522 (69.2%)
Non-autonomous repositories	2,192 (99.6%)
Broad-domain schema	10
Research workflow stages	15
Snapshot date	April 4, 2026

Phases 1–2: discovery and filtering. On April 2–3, 2026, nine GitHub code queries targeted `SKILL.md`, `CLAUDE.md`, and `.claude/` conventions; 15 repository queries searched names, descriptions, and topics. We restricted candidates to repositories active after February 24, 2025, additionally requiring creation after that date for repository searches. Adaptive file-size partitioning mitigated the 1,000-result code-search cap; repository searches retained at most 500 hits per query. Deduplication yielded 23,585 repositories, 23,569 with metadata and 21,494 with README files fetched on April 4. Weighted skill-related keyword matching on descriptions and README files retained 11,615 candidates.

Phase 3: broad domain. GPT-4o-mini assigned a domain from a ten-category menu (Appendix A.9) using descriptions, topics, and 2,000 README characters. The 331 *research* and 1,869 *data science* repositories form the 2,200-record release; data science includes applied work that is not necessarily academic. We keep data science alongside research because this category covers

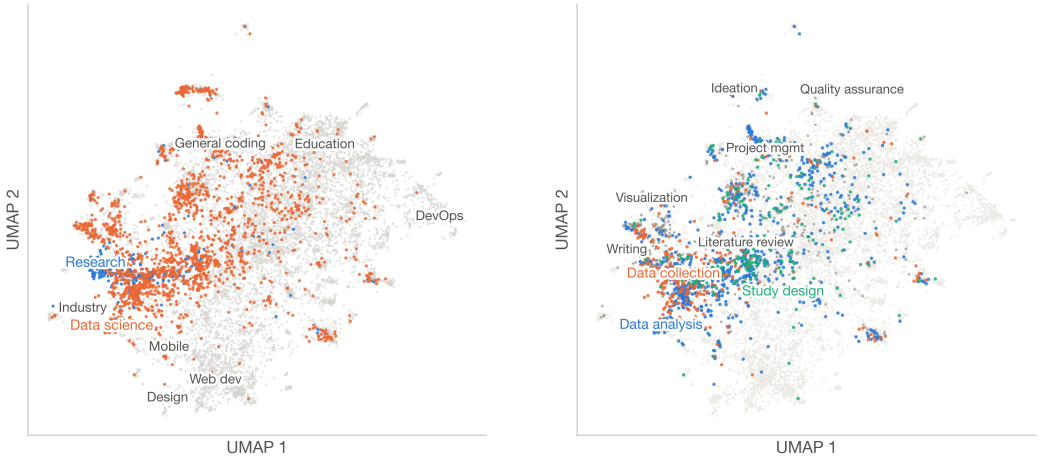


Figure 2: Embedding landscape of 11,613 valid repository summaries, projected with UMAP (cosine distance, 15 neighbors, minimum distance 0.1). Research repositories form a band inside the data-science region rather than a region of their own, and data analysis and data collection co-locate within the research-relevant corpus. (Left) Full landscape with research and data-science repositories highlighted; labels mark group density peaks. (Right) Research-relevant corpus colored by primary workflow stage. The three most frequent primary stages are highlighted, other stages are dark gray, and repositories outside the corpus are light gray. Summaries take broad domain as an input, so the layout is not independent evidence of domain structure, and axes and distances carry no quantitative meaning.

the data collection, analysis, and modeling that form the execution stages of empirical research, whereas the other eight domains serve teaching, design, industry applications, or general software development. The 21 off-menu responses were merged into *other*, without changing this subset.

Phase 4: shared summaries. GPT-4o-mini generates two-to-four-sentence summaries from repository metadata, the domain label, and up to 3,000 README characters (temperature 0.3; at most 200 output tokens; provenance in Appendix A.1). Annotation codes this summary rather than raw skill files so that every record is labeled from an input of the same form, bounded length, and snapshot date. Raw files vary much more: eight weeks on, they were retrievable for 82.0% of repositories, with median concatenated length 36,339 characters against 544 for a summary, and in a test on 161 repositories, reordering their files changes the autonomy label for 51.6% under a 12,000-character cap and 37.9% without it, against 23.0% for reordered summaries (Appendix A.6). The stored summaries serve both annotation and embedding. Of 11,615 summary fields, 11,613 contain valid summaries; the other two contain API error messages, and one of these belongs to the released corpus.

Phase 5: annotations. GPT-4o-mini jointly assigned methodology encoding, autonomy, and multi-label workflow stages at temperature 0, reading only the stored summary (median 544 characters), not full skill files. Labels for the released record whose summary is an API error message are invalid; the record stays in the release with a `record_valid` flag so that users can exclude it programmatically. A second pass received the repository name, summary, and stage tags to assign discipline and choose a primary stage within those tags; single-tag cases inherited that stage. Separately, the model assigned no stage to one other repository. As a result, 2,199 records have stage labels, 2,199 have valid summaries and embeddings, and 2,198 have both.

Phase 6: embeddings. We embedded all 11,613 valid summaries using OpenAI's `text-embedding-3-large` (3,072 dimensions) and projected them to two dimensions with UMAP (McInnes et al., 2018) (Figure 2). Labels color the plots but are not inputs to this branch.

Phase 7: human verification. The verification protocol reads the skill files alongside the summary the model was given. One author checked 100 repositories stratified by methodology and autonomy,

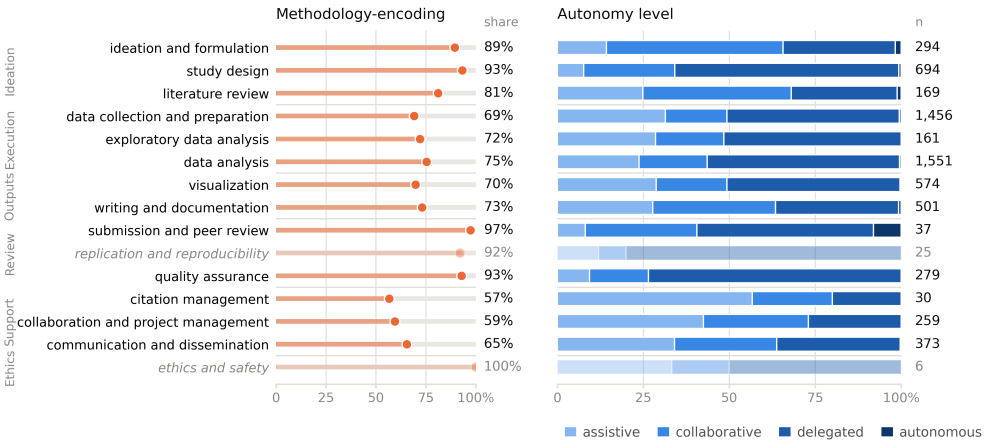


Figure 3: Methodology encoding and AI autonomy level across the 15 research workflow stages, in lifecycle order and separated by thin rules into the six lifecycle groups of Table 13, each named in the left margin in the abbreviated form Ideation / Execution / Outputs / Review / Support / Ethics. Autonomy uses one hue light to dark because its four levels are ordered. The two grayed rows rest on fewer than 30 repositories, where a single repository shifts a bar by over three points, so their percentages should not be read as structure.

with model labels visible, seeing skill names, descriptions and instruction excerpts for the 78 whose files were retrievable and the summary alone for the other 22. The author accepted 97/100 labels on each dimension; two rejections on each were the same link collections judged not applicable, leaving 98 analyzable repositories with one substantive correction on each dimension. A later check accepted 89/92 primary-stage labels on the same sample's multi-stage cases (96.7%; model–annotator $\kappa = 0.96$). These rates come from a label-shown protocol; the blinded instrument in Section 7 extends the methodology check to unanchored accuracy. Appendix A.2 reports corrections and machine-side stability tests.

5 Analysis

5.1 RQ1: Domains and Workflow Coverage

The release contains 15.0% research and 85.0% applied data-science repositories. In the released labels, research repositories more often cover literature review (33.2% versus 3.2%), submission and peer review (7.9% versus 0.6%), and citation management (6.9% versus 0.4%). Methodology encoding is also more frequent (83.1% versus 66.7%); non-autonomous shares are 98.8% and 99.8%, respectively.

Embedding the summaries places research repositories as a band within the data-science region rather than a region of their own, with data analysis and data collection co-locating in the workflow view (Figure 2). Because the summarization routine takes broad domain as an input, this layout is not independent evidence of that structure; Appendix G gives its interpretation.

Because the summaries take broad domain as an input, the research–data-science gaps above could partly reflect the domain label rather than repository content. To test this, we regenerated summaries and annotations for a stratified sample of 600 repositories with and without the domain field, using GPT-4o-mini, GPT-4o, and GPT-5, and compared the literature-review and methodology-encoding gaps (Appendix A.7, Table 3). Both gaps stay positive under every model and input. Their magnitudes vary more: the conservative 95% intervals for the change reach -23.92 and $+25.04$ points, wider than the ± 5-point tolerance we fixed before the first run, so we report the direction of both gaps and leave their magnitudes to larger samples.

Across 2,199 tagged repositories, the 15-stage taxonomy yields 6,409 assignments (2.91 per repository). Data analysis covers 70.5% of all 2,200 records and data collection and preparation 66.2%, followed by study design (31.5%) and visualization (26.1%). Execution accounts for 49.4% of

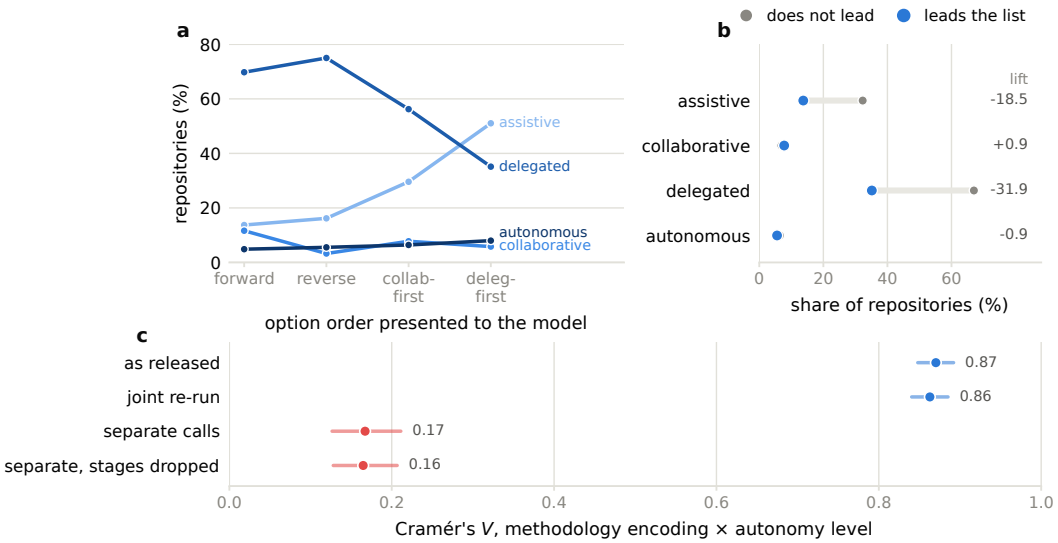


Figure 4: Two elicitation choices move the released annotations substantially. (a) Share of the 2,200 repositories at each autonomy level under four counterbalanced orderings of the option list, the prompt being otherwise identical. (b) The position effect per level, comparing a level's share when it leads the list against when it does not, so prevalence cancels and only position remains; negative lift means leading suppresses the level. (c) Cramér's V for methodology encoding against autonomy level under four elicitation structures, with 95% bootstrap intervals; the released and joint structures elicit both dimensions in one call, the separated ones independently. Panel (a) omits the released labels, which come from a three-dimension prompt and would confound order with prompt structure.

assignments and touches 87.4% of repositories. Ethics and safety (6 repositories; 0.3%), replication and reproducibility (25; 1.1%), citation management (30; 1.4%), and submission and peer review (37; 1.7%) are sparse. Stage percentages overlap (Table 13).

At least one skill file was retrieved for 1,805 repositories (82.0%) eight weeks after the snapshot. Restricting to these changes methodology encoding from 69.2% to 70.0%, leaves non-autonomous labels at 99.6%, and moves no stage by more than 1.3 points (Appendix A.4). Engagement is concentrated: the ten most-starred repositories hold 53% of stars, while 31% have none. Appendix G reports further engagement, discipline, and embedding details.

5.2 RQ2: Methodological Depth

Released labels identify 1,522 repositories (69.2%) as encoding methodological knowledge and 678 (30.8%) as procedural wrappers. Absolute prevalence is configuration-dependent: fresh domain-free weighted estimates are 52.56–58.18% for GPT-4o-mini, 47.33–48.61% for GPT-4o, and 50.15–50.21% for GPT-5, against 66.24% for the released labels on the same sample. These ranges bound missing labels, not sampling uncertainty (Appendix A.7).

Among stages with at least 30 repositories, methodology encoding is most frequent in submission and peer review (97%), study design (93%), and quality assurance (93%), and less frequent in citation management (57%), collaboration and project management (59%), and communication and dissemination (65%). Execution stages lie at 69–75% (Figure 3), where the two stages resting on fewer than 30 repositories are grayed.

5.3 RQ3: How Far the Annotations Survive Their Elicitation

Our released autonomy labels are predominantly delegated (998; 45.4%), followed by assistive (704; 32.0%), collaborative (490; 22.3%), and autonomous (8; 0.4%). The eight autonomous cases include aspirational end-to-end research systems and general-purpose agents. We now ask what this

distribution, and the association it enters, owe to the way they were elicited in the annotation process (Appendix A.8).

Option order. Re-eliciting autonomy alone under four counterbalanced orderings of the four levels changes the distribution substantially (Figure 4a). The assistive share ranges from 13.7% to 51.0% and the delegated share from 35.1% to 75.0% across orderings. Comparing each level with itself isolates position from prevalence (Figure 4b): delegated falls 31.9 points and assistive 18.5 points when the level leads the list rather than not, a mean absolute movement of 13.0 points. All four orderings agree on 1,116 of 2,200 repositories (50.7%). The autonomous share is 4.9–8.0% across these single-dimension prompts, so a non-autonomous majority persists across orderings, while its exact share depends on the prompt.

Joint versus separate elicitation. The released methodology–autonomy association is strong ($\chi^2(3) = 1666.3$, $p < 0.001$, Cramér's $V = 0.87$): methodology-encoding repositories are predominantly delegated (62.5%) or collaborative (32.1%), whereas 92.8% of procedural wrappers are assistive. Both labels came from one model call. Re-eliciting them in separate calls across all 2,200 records reduces V from 0.86 under a joint-prompt rerun to 0.17, with non-overlapping bootstrap intervals, while methodology rates remain similar (68.7% versus 68.0%) (Figure 4c). Co-elicitation, not the corpus, carries most of the association.

What this bounds. A supervised check agrees: the primary stage is recoverable from the summary the annotator read, at 0.61–0.63 accuracy against a 0.41 baseline, but stays near baseline from the skill files it never read (Appendix A.5). These perturbations hold the repositories fixed and change only the prompt, so they measure whether a label is stable, not whether it is correct. Because the autonomy distribution and its methodology association move under these perturbations, we release them as annotated fields, not ecosystem estimates (Appendix A.2).

5.4 Skill Quality

A complementary GPT-4o pilot scores 389 full `SKILL.md` files from seven purposively selected repositories on five dimensions (1–5 scale; Appendix C). Clarity (4.95) and specificity (4.91) exceed research grounding (3.59) and error prevention (3.83); composability averages 4.69 (Appendix C). Clarity and specificity reach the maximum score for 95.6% and 92.0% of files, respectively, compared with 13.1% on each of grounding and error prevention. These scores measure specification quality, not executed performance. An observational comparison with 389 length-matched non-research files has intervals spanning zero on every dimension, also with both arms on one judge snapshot, establishing neither a research advantage nor equivalence. Both sides cluster: two of its 30 repositories supply 336 files, and the design effect leaves effective sample sizes of 12.1 and 13.5 (Appendix C.1).

Composite scores also vary within and across the pilot's empirical skill types; Figure 9 and the per-type and per-repository summaries are in Appendix C.

6 Discussion

SciSkillBank makes intended research procedures inspectable. Execution-heavy coverage and frequent methodological guidance coexist with sparse ethics and replication labels. A human-in-the-loop instruction specifies intended oversight, and a risky-skill benchmark finds agents acting instead of seeking user approval (Liu et al., 2026b); instruction and behavior are separate objects, and this dataset measures the first. Section 5.3 adds that the measurement itself depends on elicitation, so the perturbation set ships with the labels.

7 Conclusion and Future Work

SciSkillBank contributes 2,200 repository records, a research-workflow codebook, annotations, summaries, and embeddings. Its focus is the corpus: what the community specifies for research agents, and how far those descriptions hold under the elicitation choices behind them. The methodology-encoding share and the non-autonomous majority each move by less than one point when the annotator

reads skill files instead of summaries (Appendix A.6). Three findings carry forward from the analysis: execution stages account for the largest share of specified work, a non-autonomous majority of repositories keeps a human in the loop, and research repositories cover literature review and encode methodology more often than applied data-science repositories. The perturbation study adds a measurement result: the methodological and human-in-the-loop shares hold across annotation inputs, while record-level labels shift with the question and the model.

The release supports uses beyond this paper. Researchers can compare how communities specify delegation across workflow stages, locate stages that receive little coverage, such as ethics and replication, or little methodological guidance, such as citation management, and select skills for curated research registries. Tool builders can also see which research tasks few repositories cover, such as submission and peer review (1.7% of repositories) and replication and reproducibility (1.1%). Researchers who study AI and work can link the stage labels to task-level measures of exposure and delegation (Eloundou et al., 2024; Shao et al., 2025). The perturbation label sets and the input-consistency test also give a reusable check on the stability of LLM annotations.

Several directions build on this foundation. The first is a human accuracy anchor for record-level use, for which we release a ready-to-run instrument: a blinded two-rater package over a fresh probability sample of 100 repositories, with full skill-file text, a coding guide, and an analysis plan. It extends the anchored methodology check to unanchored accuracy and inter-annotator agreement and adds literature-review support. The second is per-skill annotation with the codebook's skill-level dimensions (Appendix D.1), which would follow individual instructions inside a repository and complement the repository-level labels. The third is versioned release: 92.9% of these repositories still resolve 172 days on, so periodic re-crawls could track how specifications change over time and which stages gain coverage. The fourth is execution-based evaluation, which could test whether agents honor the human checkpoints that skills specify, complementing risky-skill benchmarks of unsafe agent actions (Liu et al., 2026b). Because other agents also read `SKILL.md` files, the pipeline extends to agents beyond Claude Code.

LIMITATIONS

SciSkillBank is designed to measure what public skill repositories specify for research agents, from a sample shaped by query caps and Claude Code conventions rather than a complete or cross-agent census; the corpus is 85.0% applied data science, and each domain is reported separately. Labels come from README-derived summaries, which may omit file-level instructions; skill files were retrievable for 82.0% of repositories eight weeks after the snapshot, and re-annotating from them moves the methodology share and the non-autonomous majority by less than one point (Appendix A.6). Broad domain conditions the summary routine; field-removal experiments across three models keep both contrast directions and leave their magnitudes to larger samples. Without historical request logs, the archive supports sensitivity reruns, not exact replay, and same-provider model comparisons do not establish generality across annotators. The autonomy scale's coding heuristics are not validated against an external standard, and autonomy and its association with methodology change substantially under different prompts, a finding of Section 5.3. Sparse stages rest on fewer than 30 repositories, and primary-stage choice can force distinctions between overlapping functions. Because a repository carries every stage it serves, stage groups overlap and are not independent observations for inference. One error-string record carries invalid annotations and is flagged, and a different record lacks stages.

Human verification, by one author with model labels visible, covers methodology, autonomy, and the selected primary stage, not broad domain, discipline, or every multi-label workflow assignment. Its acceptance rates are an anchored check, not unbiased accuracy or inter-annotator agreement, which the blinded two-rater instrument of Section 7 is released to measure for methodology encoding and literature-review support; a second anchored pass over 75 further repositories, weighted toward contested cases, appears in Appendix A.2. The skill-file subset check tests the stability of a statistic rather than label correctness. The purposive skill-quality pilot measures specifications, not performance, and its by-type comparisons do not separate type from repository composition or specification length. The non-research comparison depends on weighting: the error-prevention difference changes sign between equal file and equal repository weights, which target different contrasts (Appendix C.1). Stars and forks measure visibility; neither they nor the intended autonomy labels measure real use, actual oversight, adoption, scientific quality, or employment effects.

ETHICS STATEMENT

This study analyzes public GitHub repositories collected through public API endpoints, and no private repositories were included. The inputs include README content and repository metadata, including owner identifiers. Public availability does not imply an open-source license. Of the 2,200 released repositories, 957 carry no license field and a further 107 carry NOASSERTION. Every record therefore links to its source repository, and that content remains governed by the source repository's own license. The study recruited no external annotators, and one author performed the label-shown checks. Original categorization, annotation, and summarization used OpenAI GPT-4o-mini. The domain-field sensitivity experiments used `gpt-4o-mini-2024-07-18`, `gpt-4o-2024-08-06`, and `gpt-5-2025-08-07`, each for both summarization and annotation. The elicitation perturbations of Section 5.3 (Appendix A.8), the skill-file re-annotation, and the input-consistency test (Appendix A.6) used `gpt-4o-mini-2024-07-18`. The skill-quality judge in Appendices C and C.1 used `gpt-4o`, pinned to `gpt-4o-2024-08-06` for the research-arm re-scoring, and the released embeddings used OpenAI `text-embedding-3-large`.

We provide the dataset, annotations, UMAP coordinates, code, codebook, and annotation interfaces to reviewers as anonymous supplementary material, and will release them publicly, with the 3,072-dimensional embeddings, under the CC BY 4.0 license. The annotation materials include excerpts of third-party `SKILL.md` files. Before release, we replaced personal email addresses in the collected content with a placeholder. This affected one repository description and the README and skill-file evidence for five of the 100 repositories in the blinded validation sample. We kept organizational contact addresses for public databases and synthetic addresses used as documentation examples. The released materials record which files and cases this affected.

AI USE STATEMENT

We used generative AI as a tool for grammar checking, code debugging, and LLM-as-a-judge annotation. GPT-4o-mini produced the original repository categorization, skill summaries, and annotations. The domain-field sensitivity experiments used GPT-4o-mini, GPT-4o, and GPT-5 for summarization and annotation; GPT-4o also provided skill-quality judgments and a primary-stage sensitivity check. We used `text-embedding-3-large` to represent skill summaries.

AI systems did not produce the human verification labels and were not used to generate synthetic data presented as real observations. AI-generated summaries and labels are treated as model-derived measurements, and the reported sensitivity analyses examine some of their dependence on model and prompt choices.

The completed human annotation check documented in this paper involved one author reviewing methodology and autonomy labels for 100 stratified repositories and primary-stage labels for 92 multi-stage cases, and a second pass over 75 further repositories (Appendix A.2). The proposed labels were visible; skill-file evidence was available for 78 of the 100 repositories, and the remaining cases used summaries.

We take responsibility for the final content of this work, including text, claims, and implications in this manuscript.

REPRODUCIBILITY STATEMENT

The appendices document each step of the study. Appendix A covers data collection, summary provenance, annotation prompts, human verification, and sensitivity experiments, and Appendix D.1 gives the codebook. Appendix A.7 specifies the pinned models, sampling and generation settings, estimands, failure handling, and uncertainty calculations for the domain-field experiments, and Appendix A.8 gives the same details for the elicitation perturbations. Appendices C and C.1 describe the skill-quality pilot and its non-research comparison.

We provide the released materials to reviewers as anonymous supplementary material and will release them publicly after review, adding the 3,072-dimensional embeddings. They comprise repository metadata, stored summaries, annotations from both the summary and the skill-file inputs, UMAP projections, and pipeline code. They also include the perturbation label tables and code behind

Section 5.3, the script that regenerates Figure 4, and the annotation materials, including the blinded validation instrument of Section 7.

Some results cannot be reproduced exactly. With the summarization notebook, the pipeline can be rerun and extended, but the request and response logs and the exact model snapshot behind the original summaries were not kept, so a rerun will not regenerate those summaries exactly. API stochasticity and later changes to the public repositories also limit exact reproduction.

REFERENCES

AIPOCH. Medical research agent skills, 2026. URL https://github.com/aipoch/medical-research-skills. GitHub repository aipoch/medical-research-skills. Accessed: 2026-09-20.

Anthropic. Introducing our science blog, 2026. https://www.anthropic.com/research/introducing-anthropic-science.

Ruth Appel, Maxim Massenkoff, Peter McCrory, Miles McCain, Ryan Heller, Tyler Neylon, and Alex Tamkin. Anthropic economic index report: Economic primitives, 2026. https://www.anthropic.com/research/anthropic-economic-index-january-2026-report.

Joachim Baumann, Paul Röttger, Aleksandra Urman, Albert Wendsjö, Flor Miriam Plaza-del Arco, Johannes B. Gruber, and Dirk Hovy. Large language model hacking: Quantifying the hidden risks of using LLMs for text annotation, 2025. URL https://arxiv.org/abs/2509.08825.

Daniil A. Boiko, Robert MacKnight, Ben Kline, and Gabe Gomes. Autonomous chemical research with large language models. *Nature*, 624(7992):570–578, 2023. doi: 10.1038/s41586-023-06792-0.

Christian Catalini, Xiang Hui, and Jane Wu. Some simple economics of AGI. *arXiv preprint arXiv:2602.20946*, 2026.

Edward Chen. AI is threatening science jobs. Which ones are most at risk? *Nature*, 651(8104):19–20, 2026. doi: 10.1038/d41586-026-00444-9. URL https://www.nature.com/articles/d41586-026-00444-9.

Ziru Chen, Shijie Chen, Yuting Ning, Qianheng Zhang, Boshi Wang, Botao Yu, Yifei Li, Zeyi Liao, Chen Wei, Zitong Lu, Vishal Dey, Mingyi Xue, Frazier N. Baker, Benjamin Burns, Daniel Adu-Ampratwum, Xuhui Huang, Xia Ning, Song Gao, Yu Su, and Huan Sun. ScienceAgentBench: Toward rigorous assessment of language agents for data-driven scientific discovery. In *International Conference on Learning Representations (ICLR)*, 2025. https://arxiv.org/abs/2410.05080.

Simone Daniotti, Johannes Wachs, Xiangnan Feng, and Frank Neffke. Who is using AI to code? Global diffusion and impact of generative AI. *Science*, 391(6787):831–835, 2026. doi: 10.1126/science.adz9311.

Giuseppe Destefanis, Daniel Graziotin, Matteo Vaccargiu, and Marco Ortu. GitSkills: A dataset of agent skills on GitHub, 2026. URL https://arxiv.org/abs/2608.10906. To appear in Proceedings of the 24th IEEE/ACM International Conference on Mining Software Repositories (MSR '27).

Ricardo Dominguez-Olmedo, Moritz Hardt, and Celestine Mendler-Dünner. Questioning the survey responses of large language models. In *Advances in Neural Information Processing Systems*, volume 37, pp. 45850–45878. Curran Associates, Inc., 2024. doi: 10.52202/079017-1458.

Tyna Eloundou, Sam Manning, Pamela Mishkin, and Daniel Rock. GPTs are GPTs: Labor market impact potential of LLMs. *Science*, 384(6702):1306–1308, 2024. doi: 10.1126/science.adj0998.

Lukas B. Freund and Lukas F. Mann. Job transformation, specialization, and the labor market effects of AI. Institute Working Paper 114, Federal Reserve Bank of Minneapolis, September 2025. URL https://doi.org/10.21034/iwp.114. Revised March 2026.

Ruijiang Gao and Steven Chong Xiao. Nonstandard errors in AI agents. *arXiv preprint arXiv:2603.16744*, 2026.

Elizabeth Gibney. How AI slop is causing a crisis in computer science. Nature News, 2026. https://www.nature.com/articles/d41586-025-03967-9.

Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. ChatGPT outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences*, 120(30):e2305016120, 2023. doi: 10.1073/pnas.2305016120.

Juraj Gottweis, Wei-Hung Weng, Alexander Daryin, Tao Tu, Petar Sirkovic, Artiom Myaskovsky, Grzegorz Glowaty, Felix Weissenberger, Alessio Orlandi, et al. Accelerating scientific discovery with Co-Scientist. *Nature*, 655(8122):487–496, 2026. doi: 10.1038/s41586-026-10644-y. URL https://www.nature.com/articles/s41586-026-10644-y.

Mourad Gridach, Jay Nanavati, Khaldoun Zine El Abidine, Lenon Mendes, and Christina Mack. Agentic AI for scientific discovery: A survey of progress, challenges, and future directions. *arXiv preprint arXiv:2503.08979*, 2025.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Zhouchi Lin, Bowen Zhang, Lionel Ni, Wen Gao, Yuanzhuo Wang, and Jian Guo. A survey on LLM-as-a-judge. *The Innovation*, 7(6): 101253, 2026. doi: 10.1016/j.xinn.2025.101253.

Andy Hall. The 100x research institution, 2026. https://freesystems.substack.com/p/the-100x-research-institution.

Qianyue Hao, Fengli Xu, Yong Li, and James Evans. Artificial intelligence tools expand scientists' impact but contract science's focus. *Nature*, 649(8099):1237–1243, 2026. doi: 10.1038/s41586-025-09922-y.

Matthias Haslberger, Jane Gingrich, and Jasmine Bhatia. No great equalizer: Experimental evidence on productivity effects of generative AI use in the UK labor market. SSRN working paper, 2025. First posted 2023; revised 2025.

Anamaria Hell, Kateryna Vovk, Veena Krishnaraj, Jia Liu, Kosuke Aizawa, Adrian E. Bayer, Linda Blot, Jessica Cowell, Suyog Garg, Jonathan Grée, Ben Horowitz, Masaya Ichikawa, Kanyuni Iemoto, Keigo Kondo, Zacharie Lorsin, Kevin McCarthy, Jamie Robinson, Miguel Ruiz-Granda, Leander Thiele, Ievgen Vovk, and Mingshen Zhou. AI's capability in assisting scientific research in physics, astrophysics, and cosmology I: Literature review, 2026. URL https://arxiv.org/abs/2607.25672.

Fan Huang, Haewoon Kwak, and Jisun An. Is ChatGPT better than human annotators? Potential and limitations of ChatGPT in explaining implicit hate speech. In *Companion Proceedings of the ACM Web Conference 2023*, pp. 294–297, Austin, TX, USA, 2023. Association for Computing Machinery. doi: 10.1145/3543873.3587368.

Fan Huang, Haewoon Kwak, Kunwoo Park, and Jisun An. ChatGPT rates natural language explanation quality like humans: But on which scales? In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*, pp. 3111–3132, Torino, Italia, 2024a. ELRA and ICCL. doi: 10.63317/34az2vmm6ttk. URL https://aclanthology.org/2024.lrec-main.277/.

Fan Huang, Haewoon Kwak, and Jisun An. Vulnerability of LLMs' stated belief? LLMs belief resistance check through strategic persuasive conversation interventions. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 41759–41794, San Diego, California, United States, 2026a. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-acl.2074. URL https://aclanthology.org/2026.findings-acl.2074/.

Fan Huang, Songheng Zhang, Haewoon Kwak, and Jisun An. CogBias: Measuring and mitigating cognitive bias in large language models, 2026b. URL https://arxiv.org/abs/2604.01366.

Qian Huang, Jian Vora, Percy Liang, and Jure Leskovec. MLAgentBench: Evaluating language agents on machine learning experimentation. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235 of *Proceedings of Machine Learning Research*, pp. 20271–20309, 2024b. URL https://proceedings.mlr.press/v235/huang24y.html. https://arxiv.org/abs/2310.03302.

Yukun Jiang, Yage Zhang, Michael Backes, Xinyue Shen, and Yang Zhang. HarmfulSkillBench: How do harmful skills weaponize your agents?, 2026. URL https://arxiv.org/abs/2604.15415.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *International Conference on Learning Representations (ICLR)*, 2024. https://arxiv.org/abs/2310.06770.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pp. 9459–9474, 2020. URL https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html.

Lehui Li, Ruining Wang, Haochen Song, Yaoxin Mao, Tong Zhang, Yuyao Wang, Jiayi Fan, Yitong Zhang, Jieping Ye, Chengqi Zhang, and Yongshun Gong. What papers don't tell you: Recovering tacit knowledge for automated paper reproduction. *arXiv preprint arXiv:2603.01801*, 2026a.

Xiangyi Li, Yimin Liu, Wenbo Chen, Bingran You, Zonglin Di, Yifeng He, Shenghan Zheng, Kyoung Whan Choe, Jiankai Sun, Shuyi Wang, et al. SkillsBench: Benchmarking how well agent skills work across diverse tasks, 2026b. URL https://arxiv.org/abs/2602.12670.

Zhehui Liao, Maria Antoniak, Inyoung Cheong, Evie Yu-Yen Cheng, Ai-Heng Lee, Kyle Lo, Joseph Chee Chang, and Amy X. Zhang. LLMs as research tools: A large scale survey of researchers' usage and perceptions. In *Second Conference on Language Modeling (COLM)*, 2025. https://arxiv.org/abs/2411.05025.

Jiacheng Liu, Xiaohan Zhao, Xinyi Shang, and Zhiqiang Shen. Dive into Claude Code: The design space of today's and future AI agent systems, 2026a. URL https://arxiv.org/abs/2604.14228.

Qiyuan Liu, Tingfeng Hui, Kun Zhan, Kaike Zhang, and Ning Miao. OpenSkillRisk: Benchmarking agent safety when using real-world risky third-party skills, 2026b. URL https://arxiv.org/abs/2607.20121.

Chris Lu, Cong Lu, Robert Tjarko Lange, Yutaro Yamada, Shengran Hu, Jakob Foerster, David Ha, and Jeff Clune. Towards end-to-end automation of AI research. *Nature*, 651(8107):914–919, 2026. doi: 10.1038/s41586-026-10265-5.

Bodhisattwa Prasad Majumder, Harshit Surana, Dhruv Agarwal, Bhavana Dalvi Mishra, Abhijeetsingh Meena, Aryan Prakhar, Tirth Vora, Tushar Khot, Ashish Sabharwal, and Peter Clark. DiscoveryBench: Towards data-driven discovery with large language models. In *International Conference on Learning Representations (ICLR)*, 2025. https://arxiv.org/abs/2407.01725.

David Marguerit. Augmenting or automating labor? The effect of AI development on new work, employment, and wages, 2025. URL https://arxiv.org/abs/2503.19159.

Maxim Massenkoff and Peter McCrory. Labor market impacts of AI: A new measure and early evidence, 2026. https://www.anthropic.com/research/labor-market-impacts.

Leland McInnes, John Healy, and James Melville. UMAP: Uniform manifold approximation and projection for dimension reduction. *arXiv preprint arXiv:1802.03426*, 2018. URL https://arxiv.org/abs/1802.03426.

Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. State of what art? a call for multi-prompt LLM evaluation. *Transactions of the Association for Computational Linguistics*, 12:933–949, 2024. doi: 10.1162/tacl_a_00681. URL https://aclanthology.org/2024.tacl-1.52/.

Madhumita Murgia and John Burn-Murdoch. The AI job loss story is all about bundles. Financial Times, April 2026. URL https://www.ft.com/content/b69f8599-eaf1-477a-a5a8-60a715e56a04. Accessed: 2026-09-20.

Nicholas Pangakis, Samuel Wolken, and Neil Fasching. Automated annotation with generative AI requires validation. *arXiv preprint arXiv:2306.00176*, 2023.

Pouya Pezeshkpour and Estevam Hruschka. Large language models sensitivity to the order of options in multiple-choice questions. In *Findings of the Association for Computational Linguistics: NAACL 2024*, pp. 2006–2017. Association for Computational Linguistics, 2024. doi: 10.18653/v1/2024.findings-naacl.130. URL https://aclanthology.org/2024.findings-naacl.130/.

Weihong Qi, Fan Huang, Jisun An, and Haewoon Kwak. A cross-cultural comparison of LLM-based public opinion simulation: Evaluating Chinese and U.S. models on diverse societies. In *2025 IEEE International Conference on Data Mining Workshops (ICDMW)*, pp. 147–156, Washington, DC, USA, 2025. IEEE. doi: 10.1109/ICDMW69685.2025.00023.

Ariel Rosenfeld. Why I may 'hire' AI instead of a graduate student. Science, 2026. https://www.science.org/content/article/why-i-may-hire-ai-instead-graduate-student.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, volume 36, pp. 68539–68551, 2023.

Samuel Schmidgall, Yusheng Su, Ze Wang, Ximeng Sun, Jialian Wu, Xiaodong Yu, Jiang Liu, Michael Moor, Zicheng Liu, and Emad Barsoum. Agent laboratory: Using LLM agents as research assistants. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pp. 5977–6043, Suzhou, China, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-emnlp.320. URL https://aclanthology.org/2025.findings-emnlp.320/.

Kayla Schroeder and Zach Wood-Doughty. Can you trust LLM judgments? reliability of LLM-as-a-judge. *arXiv preprint arXiv:2412.12509*, 2024. URL https://arxiv.org/abs/2412.12509v2. Version 2, revised 2025.

Matthew Schwartz. Vibe physics: The AI grad student, 2026. https://www.anthropic.com/research/vibe-physics.

Melanie Sclar, Yejin Choi, Yulia Tsvetkov, and Alane Suhr. Quantifying language models' sensitivity to spurious features in prompt design or: How I learned to start worrying about prompt formatting. In *The Twelfth International Conference on Learning Representations, ICLR 2024*. OpenReview.net, 2024. URL https://arxiv.org/abs/2310.11324.

Yijia Shao, Humishka Zope, Yucheng Jiang, Jiaxin Pei, David Nguyen, Erik Brynjolfsson, and Diyi Yang. Future of work with AI agents: Auditing automation and augmentation potential across the U.S. workforce. *arXiv preprint arXiv:2506.06576*, 2025. URL https://arxiv.org/abs/2506.06576v3.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. HuggingGPT: Solving AI tasks with ChatGPT and its friends in Hugging Face. In *Advances in Neural Information Processing Systems*, volume 36, pp. 38154–38180, 2023.

Chenglei Si, Diyi Yang, and Tatsunori Hashimoto. Can LLMs generate novel research ideas? A large-scale human study with 100+ NLP researchers. In *International Conference on Learning Representations (ICLR)*, 2025. https://arxiv.org/abs/2409.04109.

Zhangde Song, Jieyu Lu, Yuanqi Du, Botao Yu, Thomas M. Pruyn, Yue Huang, Kehan Guo, et al. Evaluating large language models in scientific discovery, 2025. URL https://arxiv.org/abs/2512.15567.

Merlin Stein. How are AI agents used? Evidence from 177,000 MCP tools. *arXiv preprint arXiv:2603.23802*, 2026.

Ole Teutloff, Johanna Einsiedler, Otto Kässi, Fabian Braesemann, Pamela Mishkin, and R. Maria del Rio-Chanona. Winners and losers of generative AI: Early evidence of shifts in freelancer demand. *Journal of Economic Behavior & Organization*, 235:106845, 2025. doi: 10.1016/j.jebo.2024.106845.

Lindia Tjuatja, Valerie Chen, Tongshuang Wu, Ameet Talwalkar, and Graham Neubig. Do LLMs exhibit human-like response biases? a case study in survey design. *Transactions of the Association for Computational Linguistics*, 12:1011–1026, 2024. doi: 10.1162/tacl_a_00685. URL https://aclanthology.org/2024.tacl-1.56/.

Petter Törnberg. ChatGPT-4 outperforms experts and crowd workers in annotating political Twitter messages with zero-shot learning. *arXiv preprint arXiv:2304.06588*, 2023.

Vercel. The open agent skills ecosystem, 2026. URL https://skills.sh. Accessed: 2026-09-20.

Yanze Wang, Pengfei Yao, Tianyi Sun, Chuanrui Hu, Yan Xiao, Xiaotian Luo, Yunyun Han, Yifan Chen, Jun Sun, and Yafeng Deng. SkillCorpus: Consolidating and evaluating the open skill ecosystem for real-world LLM agents, 2026. URL https://arxiv.org/abs/2607.15557v6. Version 6.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pp. 24824–24837, 2022.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Beibin Li, Erkang Zhu, Li Jiang, Xiaoyun Zhang, Shaokun Zhang, Jiale Liu, Ahmed Hassan Awadallah, Ryen W. White, Doug Burger, and Chi Wang. AutoGen: Enabling next-gen LLM applications via multi-agent conversations. In *First Conference on Language Modeling (COLM)*, 2024. https://arxiv.org/abs/2308.08155.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations (ICLR)*, 2023. URL https://arxiv.org/abs/2210.03629.

Yongjun Zhang. Vibe researching: can AI agents with skills replace or augment social scientists? *Chinese Sociological Review*, 2026. doi: 10.1080/21620555.2026.2707167. URL https://doi.org/10.1080/21620555.2026.2707167. Advance online publication.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. Large language models are not robust multiple choice selectors. In *The Twelfth International Conference on Learning Representations, ICLR 2024*. OpenReview.net, 2024. URL https://arxiv.org/abs/2309.03882.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In *Advances in Neural Information Processing Systems (Datasets and Benchmarks Track)*, volume 36, pp. 46595–46623, 2023.

Jason Zhu. Blue book of agent skills 2026, 2026. URL https://github.com/zhuyansen/skill-blue-book/releases/tag/v1.1. AgentSkillsHub, v1.1. In Chinese. Accessed: 2026-09-20.

Caleb Ziems, William Held, Omar Shaikh, Jiaao Chen, Zhehao Zhang, and Diyi Yang. Can large language models transform computational social science? *Computational Linguistics*, 50(1): 237–291, 2024. doi: 10.1162/coli_a_00502. URL https://aclanthology.org/2024.cl-1.8/.

A TECHNICAL DETAILS

A.1 CONSTRUCTION PIPELINE

The construction pipeline (Section 4) comprises seven phases, with annotation and embedding branching from the same summaries:

1. *Repository discovery:* GitHub API code search (9 queries) + repository search (15 queries) $\rightarrow$ 23,569 candidates with metadata.
2. *README-based filtering:* README fetching via the GitHub API, paced below 5,000 requests/hour + keyword scoring $\rightarrow$ 11,615 filtered candidates.
3. *Broad-domain categorization:* GPT-4o-mini labels each of the 11,615 candidates with one broad domain, prompted from a menu of 10; retaining *research* and *data science* yields 2,200 research-relevant repositories. A keyword baseline provides a check.
4. *Summarization:* a recovered, unexecuted server-cache notebook supplies the documented GPT-4o-mini routine, including metadata, the broad-domain field, and a README excerpt. The surviving tables contain 11,613 valid summaries, 2,199 of them in the release. Historical request and response logs and the resolved model snapshot remain unavailable, as detailed below.
5. *Multidimensional annotation:* an initial pass assigns methodology encoding, autonomy, and 15-stage workflow tags from the summaries of the 2,200 research-relevant records, including one error string that was not filtered out. A second pass adds a primary stage constrained to the initial tags and a discipline.
6. *Embedding and projection:* the 11,613 valid summaries are embedded directly with `text-embedding-3-large` (3,072 dimensions) and projected with UMAP; 2,199 embeddings belong to the released corpus.
7. *Verification:* one annotator checks methodology encoding and autonomy on 100 stratified repositories and primary stage on 92 multi-stage cases (Section 4, Appendix A.2). The measures are model-versus-annotator; no inter-annotator agreement is reported.

Shared-summary provenance. The 2,200 stored summary-field values match exactly across the research-only and all-candidate summary tables and both annotation tables. The research-only summarization notebook, recovered from a server cache without saved run outputs, writes the checkpoint and specifies how each request is built. Its selected repository keys match the 2,200 archived keys exactly. The surviving all-candidate notebook reused that checkpoint and generated summaries for the remaining 9,415 candidates; its executed output records that dependency. Thus the source pipeline is runnable from the archived metadata and README evidence, while historical API logs and the resolved model snapshot remain unavailable.

A rerun therefore supports sensitivity analysis over summary content, which is how this paper uses it, but does not reproduce the exact strings the released labels and embeddings were computed from: the routine is stochastic at temperature 0.3, and the snapshot the original rolling alias resolved to is unknown.

The stored artifacts are pinned rather than merely retained: the later sensitivity runs recorded SHA-256 digests of the README checkpoint, the research-only and all-candidate summary checkpoints, the annotation checkpoint, the two source notebooks, and the derived tables. All nine verify against the archived files, and the digests agree independently across the three model manifests, which are released with the sensitivity data. The remaining provenance gap is confined to historical request and response logs and the resolved model snapshot; the summarization notebook and its request construction explain how the archived summaries were produced, and those summaries are byte-identical to what every later summary-based experiment consumed. Artifact timestamps place summarization and embedding on April 4 and initial annotation on April 5, 2026; the two branches in Figure 1 express their dependencies without implying that annotation preceded embedding.

A.2 VERIFICATION PASS

Subsample. We draw 100 of the 2,200 research-relevant repositories, stratified by the (autonomy level $\times$ methodology encoding) cell so that rare cells survive, with a floor of 12 per cell where availability allows and a fixed seed for reproducibility. All 8 *autonomous* repositories, the rarest autonomy level, are taken in full. The remaining 92 go to the six non-autonomous cells: each cell's proportional share is rounded, raised to the floor and capped at the cell's size, and the resulting excess of 20 is removed one repository at a time from the currently largest cell, which reduces delegated $\times$ yes (951 repositories) from 40 to 23 and assistive $\times$ no (629) from 26 to 23 and leaves collaborative $\times$ yes (488) at its proportional 20. Within each cell, repositories are drawn by systematic sampling across the first listed workflow-stage tag so that stage coverage is spread rather than clustered. The resulting cells are: assistive $\times$ no 23, assistive $\times$ yes 12, autonomous $\times$ yes 8, collaborative $\times$ no 2, collaborative $\times$ yes 20, delegated $\times$ no 12, delegated $\times$ yes 23. Presentation order is randomized before identifiers are assigned, so a prefix is a random subset of these selected 100, not a simple random sample of all 2,200 repositories.

What the annotator saw. Unlike the initial methodology/autonomy annotator, which received only a README-derived summary, the human annotator was shown the repository's actual `SKILL.md` files (skill names, their self-descriptions, and excerpts of their instructions) alongside the repository description and that summary. Skill files were available for 78 of the 100 repositories (1,772 `SKILL.md` files across these 78 repositories under the case-exact suffix match the sample extractor applied, median 5.5 per repository); for the remaining 22 the collection returned no `SKILL.md` (12 repositories held no matching markdown, 6 carried other Claude Code markdown, and 4 could no longer be retrieved), and the annotator worked from the summary alone. This makes the comparison different from a like-for-like replication: a disagreement can arise either because the model judged badly or because the README proxy misrepresented the repository, the limitation noted in Limitations.

Design and its bias. The pass used a verification rather than a blind design: the model's label was displayed and the annotator accepted or corrected it. Showing the proposed label may anchor the annotator and inflate agreement relative to independent coding; the size and direction of that effect were not measured. We therefore report acceptance rates and explicitly do not report them as accuracy. We also release a blinded two-rater instrument over a fresh probability sample of 100 repositories (Section 7), five of which are also in this subsample. Running it would give unanchored accuracy for methodology encoding; a separately controlled comparison could investigate anchoring.

Results. On methodology encoding the annotator accepted 97 of 100 model labels; on autonomy level, 97 of 100. Two of the three rejections on each dimension were the same two repositories, both curated collections of links to other people's skills rather than repositories that define skills of their own. The annotator marked these *not applicable* on both dimensions, a judgment about the sampling frame rather than about the labels. Excluding them leaves 98 analyzable repositories with a single substantive correction on each dimension: one repository the model called a procedural wrapper was judged to encode methodology, and one the model placed at *delegated* autonomy was judged *assistive*, a two-step move on the ordinal scale. Neither correction reverses a headline pattern. Splitting these 98 by whether skill files were available, the annotator accepted 77 of the 78 repositories with skill files and 20 of the 20 without, on each of methodology encoding and autonomy level, and, among the 90 of these that carried more than one stage tag, 69 of 72 against 18 of 18 on primary workflow stage. With 20 analyzable repositories on one side, this split has no power to detect any but a very large difference, so we report the counts without a test. These label-shown comparisons, with only 20 analyzable repositories lacking skill files, do not establish the accuracy of the summary proxy.

Machine-side stability. A complementary check perturbs the annotation rather than the annotator, on the same 92 repositories: re-wording the prompt gives $\kappa = 0.82$ (79 of 92 agree), additionally reversing the order in which candidate stages are listed $\kappa = 0.70$ (69 of 92), and re-running on GPT-4o $\kappa = 0.76$ (74 of 92). Every arm returned an on-list stage for every repository, and the primary stage falls within the repository's own workflow-stage tags in all 2,199 tagged repositories. Option-order sensitivity is therefore the dominant instability among the three perturbations tested here. A parallel check on the autonomy label, re-elicited on its own over all 2,200 repositories under four counterbalanced orderings of the four levels, finds the same sensitivity: a level's share

changes by a mean of 13.0 points according to whether it leads the list, and all four orderings agree on only 50.7% of repositories. The movement is concentrated in the two levels that dominate the distribution, *delegated* (31.9 points) and *assistive* (18.5 points), while *collaborative* and *autonomous* each move by 0.9 points. The autonomous share is 4.9–8.0% across these single-dimension prompts: a non-autonomous majority remains, but the released 99.6% is not stable across prompt designs. Those arms elicit autonomy on its own rather than jointly with the other dimensions, so they are not a drift control on the released labels. These measure whether the label is stable, not whether it is right.

Primary-stage pass. A later label-shown pass checked 92 multi-stage repositories: 89 labels were accepted (96.7%). The three primary-stage corrections were *data analysis* → *data collection and preparation*, and *writing and documentation* → *ideation and formulation* or *literature review*. Cohen's $\kappa = 0.96$ is model-versus-annotator.

Second anchored pass. A second label-shown pass was designed to test the labels where they are hardest to settle. One author judged a further 75 repositories from their skill files, none of them among the 100 above, drawn with a fixed seed (20260820) from repositories with skill files. Sixty are contested cases, where joint and separate elicitation assign different autonomy levels, stratified by the direction of the change with at least five per direction; the other 15 are cases where the two agree, stratified by the agreed level. On each contested case, the label shown was assigned at random between the two elicitations (29 joint, 31 separate). The annotator accepted 66.7% of autonomy labels and 69.3% of methodology labels, and 73.3% on each dimension among the 15 agreement cases. Because the sample concentrates on contested cases by design, these rates describe the hardest labels and are not directly comparable with the 97% of the first pass. On the contested cases, the annotator accepted the jointly and the separately elicited autonomy labels at 65.5% and 64.5%, a 1.0-point difference with overlapping intervals and within the 15-point margin fixed before scoring, so the human check does not separate the two elicitations.

What this does not establish. With one human annotator, we report no human inter-annotator κ; 0.96 above compares model and annotator. Anchored acceptance is an informed check, not unbiased accuracy or reliability, and the codebook's target of a κ of at least 0.70 (Appendix D.1.5) remains untested. The two *not applicable* cases motivate a separate prevalence audit of aggregators.

A.3 GitHub Search Limitations

GitHub's code search API returns a maximum of 1,000 results per query regardless of pagination. A probe of our base queries showed total counts of 62,080 (`SKILL.md` in `.claude/`), 148,736 (`SKILL.md` in `skills/`), 96,224 (`SKILL.md` anywhere), 32,560 (`CLAUDE.md`), and 7,992 (`.claude/settings.json`). The 23,569 repositories form a non-exhaustive candidate set, including projects that do not define agent skills. Query totals count overlapping file hits and may include non-agent uses of similarly named files and forked copies; they are not population counts of unique skill repositories. This redundancy is a general property of the ecosystem: SkillCorpus (Wang et al., 2026) reports that an exact content- and name-fingerprint tier alone collapses 169,465 duplicate skill files, 59.7% of the files entering its deduplication stage.

A.4 Robustness of the Headline Statistics

Two robustness questions bear on the headline statistics: whether they hold across the research and applied data-science strata, and whether they hold on the subset that still carries skill files.

Across strata, Section 5.1 gives the non-autonomous share and the methodology-encoding rate, which move in opposite ways: the first barely at all, the second by 16 points. The autonomy distribution shifts as well, in a way that is not visible in either figure: among the 331 research repositories *collaborative* (25.1%) outranks *assistive* (20.5%), inverting the pooled ranking of those two levels, a ranking that holds among the 1,869 data-science ones.

Collecting skill files for all 2,200 released repositories on May 31 and June 1, 2026, eight weeks after the snapshot, returned at least one file named `SKILL.md`, matched case-insensitively as the collector does, for 1,805 of them (82.0%); of the remainder, 102 carry other Claude Code markdown (87 a `CLAUDE.md`), 209 were reachable but held no matching file, and 84 could no longer be retrieved. The

gap tracks one discovery route: 94.0% of the 1,351 repositories found by code search alone retain a skill file, against 60.5% of the 795 found by repository keyword search alone, which matches names, descriptions, and topics rather than file contents.

Restricting the corpus to those 1,805 moves the headline statistics very little: methodology encoding goes from 69.2% to 70.0%, non-autonomous repositories remain at 99.6%, the applied data-science share moves from 85.0% to 84.2%, and no workflow stage moves by more than 1.3 points. Some secondary descriptives move further without changing any reading: the *delegated* share rises from 45.4% to 48.1% overall and from 62.5% to 65.4% within methodology-encoding repositories, where the *collaborative* share correspondingly falls from 32.1% to 29.7%; across the whole corpus the computer-science share falls from 74.7% to 72.4%, the top-10 share of forks falls from 61% to 57%, and the zero-star share rises from 31% to 33%. These discipline and engagement statistics are also sensitive to the subset; the autonomy movements shift slightly toward delegation while leaving the non-autonomous share, on which the human-in-the-loop claim rests, at 99.6%. Finer cross-tabulations, such as autonomy level within a single workflow stage, move by up to 5 points among stages carrying at least 30 repositories, and further on the sparse stages flagged in the Limitations.

One completeness detail belongs with these: the repository carrying no embedding is not the one left without a workflow-stage label. Its summarization call failed, the error message was stored in the summary field, and that repository's labels were produced from that string rather than from a description of its contents.

The codebook in Appendix D.1 specifies full `SKILL.md` content as the input for a per-skill coding pass; the repository-level pipeline codes the shared summary, which gives every repository an input of the same form and bounded length (Section 4). Appendix A.6 re-annotates the corpus from skill-file content to measure what that choice changes. The pilot study of Appendix C complements it by assessing full skill-file specification quality in a curated subset; its different sample and rubric do not independently validate the corpus labels. Prior work finds strong performance on specific text-annotation tasks (Gilardi et al., 2023; Törnberg, 2023); the blinded instrument described in Section 7 would further strengthen the released labels.

A.5 Label Recoverability from Skill-File Content

The released annotations support a concrete supervised task, and running it measures something the rest of the paper can only argue: how far a released label is carried by the evidence the annotator read rather than by the repository it describes.

Task. Predict a repository's released `primary_stage` from its text. Two conditions use the same repositories, the same folds, and the same model, and differ only in which text they see. The *summary* condition uses the README-derived summary the annotating model actually read. The *skill-file* condition uses the concatenated `SKILL.md` files, which the annotating model never saw: they were collected in late May and early June 2026, whereas the labels were assigned in April from the summary alone. The skill-file condition is therefore held-out evidence with respect to the label rather than a re-reading of its own input.

Data and protocol. Of the released repositories, 1,772 carry both a primary stage and at least one skill file under a case-sensitive file-name match (median 6 files and 36,339 characters per repository). Forks and template copies are common in this ecosystem, so we group repositories by near-duplicate skill-file content and hold every group entirely within one fold. Grouping unions exact content hashes with pairs whose sampled word 5-gram shingles overlap at Jaccard 0.8 or above, candidates being drawn from shared sampled shingles rather than compared exhaustively; the sampling makes the procedure conservative, merging fewer repositories than an exhaustive comparison at the same threshold would. The grouping script is released with the code. This collapses 1,772 repositories into 1,736 groups, the largest holding six copies of one curated list. We use stratified group-aware five-fold cross-validation, TF-IDF over word unigrams and bigrams, and multinomial logistic regression. One class, *ethics and safety*, has a single repository here and cannot be evaluated at five folds; it is excluded, leaving 14 classes and 1,771 repositories, the smallest retained class holding five. The baseline predicts the majority class, which covers 40.7%.

Table 2: Recovering the released primary stage from repository text, over 1,771 repositories in 1,735 near-duplicate groups and 14 classes, under stratified group-aware five-fold cross-validation. The summary condition uses the text the annotating model read; the skill-file condition uses content it never saw. Fold standard deviations are at most 0.043 on accuracy and 0.033 on macro F_1.

	Balanced weights		Unweighted	
Condition	Accuracy	Macro F_1	Accuracy	Macro F_1
Majority-class baseline	0.407	0.041	0.407	0.041
Summary, read by the annotator	0.629	0.483	0.607	0.296
Skill files, not read by the annotator	0.344	0.194	0.421	0.088

Results. Table 2 reports both conditions under balanced and unbalanced class weighting, since the two weightings trade accuracy against macro-averaged F_1.

The summary condition is well clear of the baseline under both weightings, reaching 0.607 to 0.629 accuracy and a macro F_1 seven to twelve times the baseline. The skill-file condition is far weaker: its accuracy of 0.421 sits within one fold standard deviation of the 0.407 baseline without class weights and falls below it, at 0.344, with balanced weights, and its macro F_1 of 0.088 to 0.194, while two to five times the baseline, indicates only a faint signal. The direction of this gap is stable across both weighting choices.

Reading. The labels are substantially predictable from the summary and only faintly predictable from the repository content that summary describes. This measures, on 1,771 repositories rather than a stratified subsample, a question the label-shown human passes could not settle: whether what these annotations express is available in the README-derived summary rather than in the skill files. It also gives the direction of the improvement we identify in Section 7: an annotation pass that reads skill files would be predicting from evidence that a summary-trained label does not presently reflect.

Two limits bound the reading. TF-IDF with a linear classifier is a deliberately simple probe, so these figures bound recoverability from below and a stronger model could narrow the gap. And the skill files postdate the labels by eight weeks, so some of the gap reflects repository change rather than information loss in the summary. The task, the grouping procedure, the folds and the code are released, so both bounds can be tightened by others.

A.6 Re-annotation from Skill-File Content

Appendix A.5 asks whether the released label can be predicted from skill-file content. This asks the complementary question directly: what label does the annotator assign when it reads the skill files instead of the summary?

Design. We re-ran the Phase 5 annotation over 1,772 of the 1,805 released repositories with retrievable skill files, the extraction here requiring a case-sensitive file name and a primary-stage label, holding the model (`gpt-4o-mini-2024-07-18`, the pinned form of the family that produced the released labels), the temperature, and the prompt fixed, and changing only the input: the concatenated `SKILL.md` text, capped at 12,000 characters, in place of the README-derived summary. Every call returned parseable JSON with an on-menu methodology and autonomy label; 14 responses also carried at least one off-menu workflow stage, counted in the overlap below but dropped from the released table, where 27 records carry no skill-file stage. The run cost 0.95 USD.

Results. The per-repository labels move substantially while the corpus-level shares do not. Agreement with the released labels is 69.2% (1,226 of 1,772) on methodology encoding and 49.7% (881 of 1,772) on autonomy level, and the multi-label workflow-stage sets overlap at a mean Jaccard of 0.358. Yet the methodology-encoding share is 69.9% on the released labels and 70.8% when the model reads the skill files, and the non-autonomous share is 99.6% against 99.3%. Within the autonomy scale the composition shifts: *collaborative* falls from 20.7% to 5.6% and *delegated* rises from 48.0% to 59.7%, while *autonomous* is 0.4% and 0.7%.

Reading. Two conclusions follow, and they point in opposite directions, which is why we report both. The corpus-level prevalence figures barely change when the annotator reads the artifact itself rather than a summary of it: methodology encoding and the non-autonomous majority are almost unchanged. They are not stable across models or prompts, however (Sections 5.2 and 5.3). An individual record's label is far less durable, since 50.3% (891) of the autonomy assignments and 30.8% (546) of the methodology assignments change, and the stage sets overlap at a mean Jaccard of 0.358. Read together with Section 5.3, this supports using the released annotations for aggregate description under a stated configuration, while treating a single repository's label as provisional. It also qualifies the ordering of the autonomy levels specifically: the balance between *collaborative* and *delegated* depends on whether the annotator sees the instructions or a description of them.

Repository persistence. We also re-queried every released repository against the public GitHub repository endpoint on September 23, 2026, 172 days after the April 4 snapshot. Of the 2,200, 2,044 resolve (92.9%) and 156 return not-found; 1,960 (89.1%) resolve under the exact released identifier, the remaining 84 only through a rename or transfer redirect, and 48 of the survivors are archived. We also re-checked skill-file presence itself, giving a second point on the curve the paper previously had only one reading of: 78.7% of the corpus still exposes a `SKILL.md` at 172 days, against 82.0% at eight weeks, and 95.2% of the 1,805 repositories that had one at eight weeks still do. Skill-file presence at the snapshot itself was not recorded, so the 18.0% without a skill file at eight weeks cannot be read as attrition alone.

Both label sets are released. Each record with retrievable skill files carries its skill-file annotation beside the summary-derived one, so a user who judges the summary proxy too weak for a particular question can work from instructions instead, and the two can be compared per record rather than only in aggregate.

The comparison shares the limits of its inputs. The skill files postdate the labels by eight weeks, the 12,000-character cap truncates 74.5% of these repositories, whose median skill text, 36,339 characters, is three times the cap, and both arms are the same model family, so this measures input substitution rather than annotator accuracy. For the same reasons the two inputs are not like-for-like: the skill-file input differs in date, in coverage, and, for truncated repositories, in which files the annotator sees at all, so a per-record disagreement does not by itself locate an error in the summary-derived label.

Input consistency. A separate test asks which input yields the more consistent label. We drew 200 of the 1,770 released repositories that carry a file named exactly `SKILL.md` and a valid annotated record (the 1,772 of Appendix A.5 less the record whose summary is an API error message and one repository whose skill files match only by suffix, such as `adr.SKILL.md`) and annotated each with the Phase 5 prompt, `gpt-4o-mini-2024-07-18` and temperature 0 under three inputs: the stored summary, the concatenated skill files cut at 12,000 characters as above, and the same files cut only at 360,000 characters. Each input was presented in three orderings of its own units, sentences for the summary and files for the skill text, so the packaging varied while the content did not, except where a cap decides which files survive. Among the 161 repositories where both inputs have at least two units, the autonomy label is identical under all three orderings for 77.0% of repositories from the summary, 48.4% from the capped skill files and 62.1% from the uncapped ones. Methodology encoding is identical for 83.9%, 66.5% and 76.4%, and the stage sets overlap at a mean pairwise Jaccard of 0.829, 0.514 and 0.605. Paired against the summary, the capped input is less stable on all three (autonomy -28.6 points, 95% repository bootstrap CI $[-39.1, -18.0]$; methodology -17.4, $[-26.7, -8.1]$; Jaccard -0.316, $[-0.366, -0.264]$), and the uncapped input on autonomy (-14.9, $[-25.5, -4.3]$) and stage overlap (-0.225, $[-0.272, -0.178]$) but not detectably on methodology (-7.5, $[-16.1, +1.2]$). The 21 repositories whose skill text fits within the cap show no detectable difference, at a size that could reveal only a large one. Length varies in the same direction: across the 1,770 repositories, the 90th percentile of skill-text length is 60.8 times the 10th, against 1.3 for summaries. At canonical order the summary arm reproduces 97.0% of the released methodology labels and 96.0% of the autonomy labels, so it matches the released pipeline. The raw-file input is thus the less stable of the two, and part of the per-record disagreement above reflects that instability rather than error in the summary-derived label. The run cost 2.58 USD.

A.7 Domain-Field Sensitivity across Models

Design. We regenerated summaries and initial annotations for the same 600 repositories with and without the complete `Category:` line in the summarizer input. We first ran `gpt-4o-2024-08-06`, then added `gpt-4o-mini-2024-07-18` to test the original model family and `gpt-5-2025-08-07` to examine transfer to a newer configuration. Each model supplies both stages in both fresh conditions, using the same archived metadata, README excerpts, literal prompt strings, sample, and repetition subset. GPT-4o and GPT-4o-mini use summarization temperature 0.3 with at most 200 output tokens and annotation temperature 0 with at most 300. GPT-5 uses its supported `minimal` reasoning setting, omits temperature, and allows 1,024 completion tokens per stage, including reasoning and visible output; these compatibility changes limit attribution of cross-model differences to model identity alone. For every model, annotation receives at most 2,000 characters of the generated summary, and settings are fixed across the two domain-field conditions. The intervention removes only the explicit field; domain cues in the repository evidence remain. The valid-summary frame excludes the archived API-error record and contains 331 research and 1,868 data-science repositories. Using seed 20260919, we sampled without replacement within four strata defined by broad domain and the archived literature-review tag: 110/110 research positives, 190/221 research negatives, 59/59 data-science positives, and 241/1,809 data-science negatives (sample/frame sizes). Each domain therefore contributes 300 repositories, with stratum expansion weights N_h/n_h for estimation. The two follow-ups were selected after observing the GPT-4o results; they retain the initial outcomes, sampling design, failure policy, and practical margins rather than constituting newly outcome-blind experiments.

Estimands and missing responses. For each outcome, let G_c denote the weighted research-minus-data-science gap in condition c; the paired intervention contrast is $\Delta = G_{\text{without}} - G_{\text{with}}$. We fixed a practical tolerance of ± 5 percentage points for each outcome before the first GPT-4o generation and retained it in both follow-ups. The joint-schema validator rejects an annotation if any requested categorical field is invalid; it does not map off-menu workflow labels to allowed ones. All 1,200 primary summaries per model were valid. After at most three attempts, 42, 22, and 3 primary condition annotations remained invalid under GPT-4o-mini, GPT-4o, and GPT-5, respectively, leaving 564, 580, and 597 complete repository pairs. All unresolved annotations used off-menu workflow stages; they were not truncated responses. We retained the original sampling weights and bounded missing binary outcomes by 0 and 1 rather than dropping repositories or substituting complete-case estimates. Table 3 gives the resulting ranges of possible full-sample weighted gaps. Conservative 95% bounds additionally propagate simultaneous exact hypergeometric intervals for the non-census strata and directional transitions, allowing every assignment to missing outcomes. They describe finite-corpus sampling uncertainty conditional on generated outputs, not annotation accuracy or all generation variability; they are not adjusted across the three model configurations.

Results and scope. Both domain contrasts remain positive in each configuration. For GPT-4o-mini, GPT-4o, and GPT-5, respectively, conservative 95% bounds for the domain-free literature-review gap are [21.76, 36.84], [29.35, 37.27], and [38.35, 52.61] percentage points; the corresponding methodology-encoding bounds are [17.54, 41.07], [16.99, 35.98], and [20.64, 37.25]. However, no interval for Δ lies wholly within the retained ± 5-point tolerance. The contrasts therefore keep their direction under these configurations, a weaker criterion than the pre-committed tolerance, while their magnitudes remain unresolved. Absolute methodology rates vary substantially: the archived weighted sample estimate is 66.24%, compared with domain-free estimator ranges of 52.56–58.18%, 47.33–48.61%, and 50.15–50.21% for mini, 4o, and GPT-5. The fresh with-domain ranges are also lower (53.26–59.96%, 43.16–46.78%, and 47.11–47.80%), so the archived-to-fresh decline cannot be attributed to explicit field removal alone. These prevalence ranges describe missing-label ambiguity in weighted sample estimates, not confidence intervals. Archived comparisons combine regeneration and incomplete historical request provenance with model substitution where applicable; GPT-5 also changes supported generation settings. Parsing also differs: the archived annotator filtered off-menu stage values, whereas the fresh experiments reject an invalid joint schema in full. The mini run tests the original model family without reproducing unknown historical requests, and the GPT-5 run broadens the configuration check without establishing model superiority or accuracy against repository contents.

Table 3: Paired domain-field sensitivity on the same 600 repositories. All quantities are in percentage points (research minus data science); Δ is without minus with. The two notations are distinct and not interchangeable. A dashed range a–b is an identification range: it bounds the weighted sample estimator over the binary labels left unresolved after three attempts, and is not a confidence interval. A bracketed $[a, b]$ is a conservative 95% bound incorporating missing outcomes and finite-population sampling, conditional on generated outputs. Prevalence ranges quoted in the text use the dashed notation for the same reason. Bounds are rounded outward. Models are pinned snapshots used for both summarization and annotation; GPT-5 uses different supported generation settings. Follow-ups were selected after the GPT-4o experiment, and intervals are not adjusted across models.

Outcome	Model	With-domain gap	Domain-free gap	Δ: 95% bounds
Literature review	GPT-4o-mini	25.25–33.71	27.08–34.28	$[-20.29, 22.24]$
	GPT-4o	32.11–36.95	33.01–35.38	$[-14.15, 10.31]$
	GPT-5	42.30–43.12	45.87–45.93	$[-6.65, 12.08]$
Methodology encoding	GPT-4o-mini	23.66–32.12	25.93–33.12	$[-23.92, 25.04]$
	GPT-4o	25.32–30.16	25.54–27.92	$[-17.67, 16.13]$
	GPT-5	30.32–31.13	29.21–29.27	$[-12.21, 8.70]$

Repeated generation and diagnostic audit. Forty fixed repositories, ten per sampling stratum, received one additional paired repetition under each model, yielding 39, 39, and 40 complete repeated pairs for mini, 4o, and GPT-5. Among valid annotations available in both runs, literature-review labels agreed in 36/40 and 35/39 mini cases, 33/38 and 37/40 GPT-4o cases, and 35/40 and 36/40 GPT-5 cases, respectively for with-domain and without-domain conditions. Methodology agreement was 39/40 and 34/39 for mini, 30/38 and 29/40 for GPT-4o, and 32/40 and 35/40 for GPT-5. These unweighted diagnostics concern the enriched repetition subset, not additional independent repositories or corpus-wide agreement estimates; they show generation variability beyond the sampling uncertainty above. An unblinded diagnostic inspection of all 22 unresolved GPT-4o conditions found that 21 repeated the same off-menu label set across all three attempts. The summaries and rejected labels share activity terms such as coding and deployment that the summarizer's examples include but the closed annotation menu omits. A seeded inspection of methodology changes also found both altered substantive detail and differing judgments of similar descriptions. These observations identify possible sources of instability, without rescuing invalid labels, isolating a causal mechanism, or constituting human accuracy validation.

A.8 Elicitation Perturbations

Model and settings. Both perturbation runs in Section 5.3 requested the pinned snapshot `gpt-4o-mini-2024-07-18`, which the API reported for all 17,100 billed calls, and ran on August 20, 2026. Calls used temperature 0, JSON response mode, and a single user message, with at most 160 output tokens for the joint-versus-separate run and 120 for the option-order run. Responses from a 300-repository pilot, requested under the same identifier, were reused. The only per-repository input was the stored Phase 4 summary from which the released labels were coded (median 544 characters, maximum 761), for all 2,200 repositories, including the record whose summary is an API error message. No call failed, so every structure and ordering has a label for all 2,200 repositories.

Joint versus separate elicitation. Figure 4c compares four structures. *As released* are the Phase 5 labels, produced through the rolling `gpt-4o-mini` alias with no recorded snapshot. The *joint rerun* sends the same three-dimension prompt (Appendix A.9) to the pinned model, with five dashes and two arrows typed as ASCII. The *separate* structure asks for methodology and autonomy in two calls per repository, each keeping the workflow-stage block and dropping the other dimension's block and example field; *separate without stages* asks for a single dimension per call. Definitions, the methodology heuristic, and the order of the levels are unchanged across structures.

Option order. The option-order run asks for autonomy alone, using the autonomy definitions of the joint prompt with the four levels listed in four orders: assistive, collaborative, delegated, autonomous; the reverse; collaborative, autonomous, assistive, delegated; and delegated, assistive, collaborative, autonomous. Each level leads exactly once, and only the order of the definitions changes.

Uncertainty. Intervals for Cramér's V in Figure 4c are 95% percentile intervals from 5,000 bootstrap resamples of repositories (seed 20260923), computed for each structure separately; the archived run reports, which used 2,000 resamples, differ from them by at most 0.002. Option-order agreement and position effects are reported as descriptive statistics.

A.9 LLM PROMPTS

Four model instructions are recorded for the pipeline: categorization (Phase 3), summarization (Phase 4), and the initial and primary-stage/discipline annotation passes (both Phase 5). The Phase 3 entry is an adapted, domain-only reconstruction; the summarization instruction is preserved in the summarization notebook, subject to the historical request-log and model-snapshot limits in Appendix A.1; the two annotation instructions reproduce the executed prompts subject to the omissions stated below. The per-repository inputs were metadata, broad-domain label, and README excerpt for summarization; the summary for initial annotation; and the repository name, summary, and initial stage tags for the second annotation pass.

Domain-categorization instruction (Phase 3). The following domain-only reconstruction preserves the inputs and ten broad-domain definitions of the executed pass while expressing only the output retained in the released dataset. It is adapted to the released schema and is not a verbatim transcript of the complete system message.

```
You are categorizing a GitHub repository that contains Claude Code
configuration. Based on the repo info and README excerpt, assign the
repository's broad domain.
Broad domain: Choose ONE from: "research" - academic/scientific
research, "data_science" - data science, ML, AI (non-academic),
"education" - teaching, tutorials, courses, "design" - UI/UX, frontend
design, "industry" - enterprise, SaaS, business tools, "devops" -
infrastructure, CI/CD, cloud, "web_dev" - web application development,
"mobile" - mobile app development, "general_coding" - general software
development, "other" - doesn't fit any above.
Return ONLY valid JSON: {"broad_domain": "...", "reason": "one sentence"}
```

Summarization prompt (Phase 4). The summarization notebook's instruction for generating the shared summaries consumed by both annotation passes in Phase 5 and embedded in Phase 6:

```
You are summarizing a GitHub repository's skills and capabilities for a
research study. Based on the repo info and README, produce a concise
summary (2-4 sentences) covering: 1. What skills or capabilities the
repo provides. 2. What workflow stage(s) it supports (e.g., coding, data
analysis, writing, deployment). 3. Any notable tools, frameworks, or
methodologies it references. Please be factual and specific. Focus on
the SKILLS, not the repo's popularity.
```

Initial annotation prompt (Phase 5). Used to produce the methodology-encoding, autonomy-level, and workflow-stage labels from the skill summary. Reproduced as run, less the enumeration of the 15 stage identifiers, which the prompt lists in `snake_case` in the order of the list in Appendix D.1.2:

```
You are annotating a GitHub repository that contains AI research skills
(Claude Code skill files). Based on the repository's skill summary
below, classify it along THREE dimensions.
Dimension 1 - Methodology Encoding (binary): Does this repository's
skills encode tacit methodological judgment and research best practices,
or are they procedural tool wrappers? "yes": Skills capture tacit
knowledge - research best practices, decision frameworks,
domain-specific methodological guidance (e.g., "apply Bonferroni
correction", "use PRISMA checklist", "GATK best practices"). "no":
Skills are procedural wrappers - they automate mechanical tasks without
embedding methodological judgment (e.g., "format this table", "run this
CLI command", "convert citation styles"). Heuristic: if you removed the
skill's instructions and gave the LLM only the task name, would the
```

```
output miss important methodological guardrails? If yes → "yes". If
output would be roughly the same → "no".

Dimension 2 — AI Autonomy Level (single label): How much independent
decision-making do the skills assume for the AI agent? "assistive":
Human decides what and how; AI handles mechanics. "collaborative": AI
proposes options/drafts; human selects and validates. "delegated": AI
executes multi-step workflows with judgment calls; human sets goal and
reviews output. "autonomous": AI identifies what to do, executes, and
self-validates; human involvement optional. Code the INTENDED autonomy
from the skill instructions, not the LLM's raw capability.

Dimension 3 — Workflow Stages (multi-label): Which research workflow
stages does this repository target? Choose ALL that apply from: [the 15
stage identifiers]

Return ONLY valid JSON, for example: {"methodology_encoding": "yes",
"autonomy_level": "collaborative", "workflow_stages": ["data_analysis",
"visualization"], "reason": "one sentence"}

SKILL SUMMARY: {summary}
```

Primary-stage and discipline prompt (Phase 5). Used for the second annotation pass, which produces the single-label `primary_stage` and the repository-level `discipline`. It receives the same skill summary and repository name; the primary stage is constrained to the initial pass's tags, interpolated at `{tags}`:

```
You are annotating a GitHub repository that contains AI research skills
(Claude Code skill files). Based on the repository's skill summary
below, assign TWO labels.

Label 1 - Primary workflow stage (single label). The repository has
already been tagged with these research workflow stages: {tags} Choose
the ONE that is the repository's primary purpose - the stage a researcher
would go to this repository FOR. You must choose from that list and
nothing else. If the repository serves several equally, choose the most
specific rather than the broadest.

Label 2 - Discipline (single label). Which research discipline does this
repository primarily serve? "life_sciences": genomics, bioinformatics,
ecology, neuroscience, medicine, biology. "physical_sciences": physics,
chemistry, materials science, astronomy, earth science.
"social_sciences": economics, political science, psychology, sociology,
education. "finance_business": quantitative finance, markets, business
analytics. "regulatory_legal": compliance, legal analysis, regulatory
submissions. "computer_science": ML/AI research, systems, software
engineering research. "cross_domain": general-purpose research tooling
not specific to any field. Choose "cross_domain" when the repository is
a general research toolkit. Do not guess a field from incidental example
data; judge what the skills are BUILT for.

Repository: {repo} Skill summary: {summary}

Respond with ONLY a JSON object: {"primary_stage": "<one of the tagged
stages>", "discipline": "<one discipline>", "discipline_confidence":
"<high|medium|low>"}
```

A.10 Most-Starred Repository per Workflow Stage

Table 4 lists the most-starred repository carrying each research workflow stage. These rows are illustrative of visibility rather than of research function, and several make that distinction visible: the most-starred repository at five stages is a general-purpose Claude Code collection, three of these rows being the same repository, and at two it is an archive of extracted AI-tool system prompts. Two properties produce this. Stars are extremely concentrated (the ten most-starred repositories hold 53% of all stars), so a stage's maximum is usually drawn from a small set of very popular general collections; and because stage tagging is multi-label, one broad collection can head several stages at once. Read as a face-validity check on the labels rather than as examples, the table shows the limit of README-derived stage assignment measured in Appendix A.6: a repository whose

Table 4: The most-starred repository carrying each research workflow stage. A repository is tagged with every stage it serves, so the same repository can head more than one row.

Workflow stage	Most-starred repository	Stars
ideation and formulation	NevaMind-AI/memU	13,287
study design	anthropics/claude-cookbooks	37,233
literature review	Orchestra-Research/AI-Research-SKILLs	6,038
data collection and preparation	affaan-m/everything-claude-code	134,326
exploratory data analysis	asgeirtj/system_prompts_leaks	36,766
data analysis	affaan-m/everything-claude-code	134,326
visualization	gsd-build/get-shit-done	47,207
writing and documentation	hesreallyhim/awesome-claude-code	36,019
submission and peer review	Imbad0202/academic-research-skills	1,842
replication and reproducibility	ZimoLiao/scholaraio	258
quality assurance	affaan-m/everything-claude-code	134,326
citation management	Galaxy-Dawn/claude-scholar	3,059
collaboration and project management	x1xhlol/system-prompts-and-models-of-ai-tools	134,193
communication and dissemination	ComposioHQ/awesome-claude-skills	50,685
ethics and safety	MaxwellCalkin/sentinel-ai	9

README advertises analysis, visualization, and quality assurance receives those tags whether or not a researcher would reach for it at those stages.

B Supplementary Figures and Tables

Table 5 reports the full fine-grained workflow-stage distribution summarized in Section 5.1. This annotation is close to exhaustive: only one repository received no stage label.

Table 5: Fine-grained workflow-stage distribution over the 2,200 research-relevant repositories (Phase 5 annotation, multi-label). Percentages are of all 2,200 repositories and do not sum to 100% because a repository is tagged with every applicable stage (6,409 assignments, mean 2.91 per repository). 2,199 of 2,200 repositories received at least one stage.

Workflow stage	Repos	%
data analysis	1,551	70.5
data collection and preparation	1,456	66.2
study design	694	31.5
visualization	574	26.1
writing and documentation	501	22.8
communication and dissemination	373	17.0
ideation and formulation	294	13.4
quality assurance	279	12.7
collaboration and project management	259	11.8
literature review	169	7.7
exploratory data analysis	161	7.3
submission and peer review	37	1.7
citation management	30	1.4
replication and reproducibility	25	1.1
ethics and safety	6	0.3

Beyond the analyses in the main text, several figures further characterize the research-relevant repositories. Figure 6 details the per-stage stars/forks distributions for the long tail discussed in Section 5.1. Figure 7 reports the programming-language distribution (Python dominates, followed by repositories with no language set, then TypeScript and Shell), and Figure 8 reports repository size across stages.

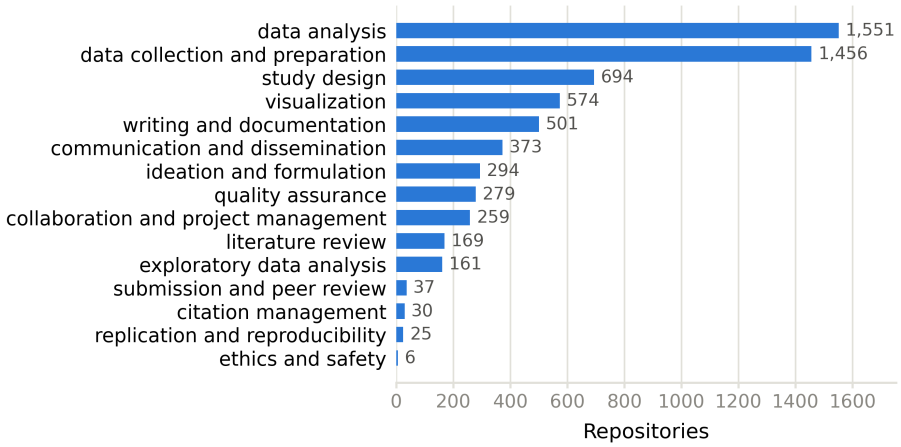
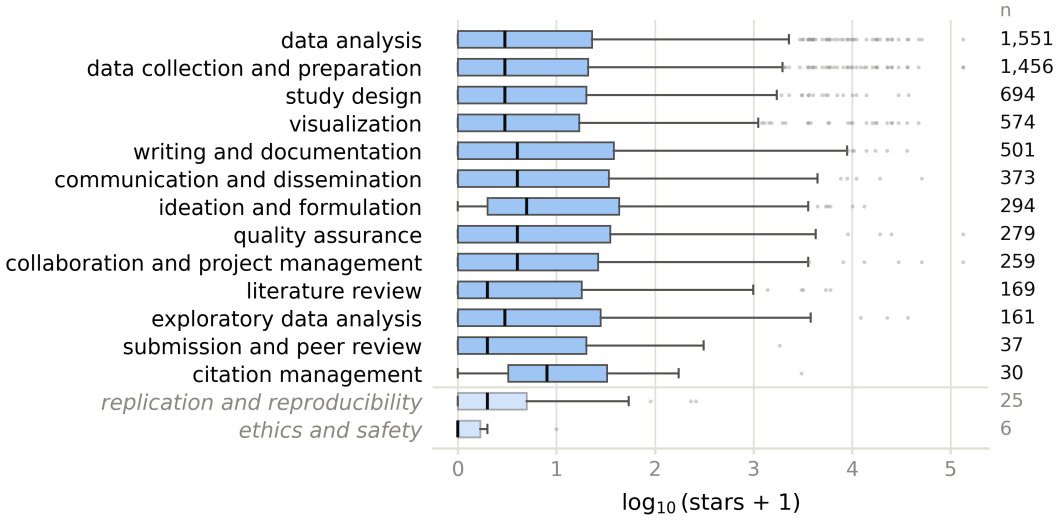
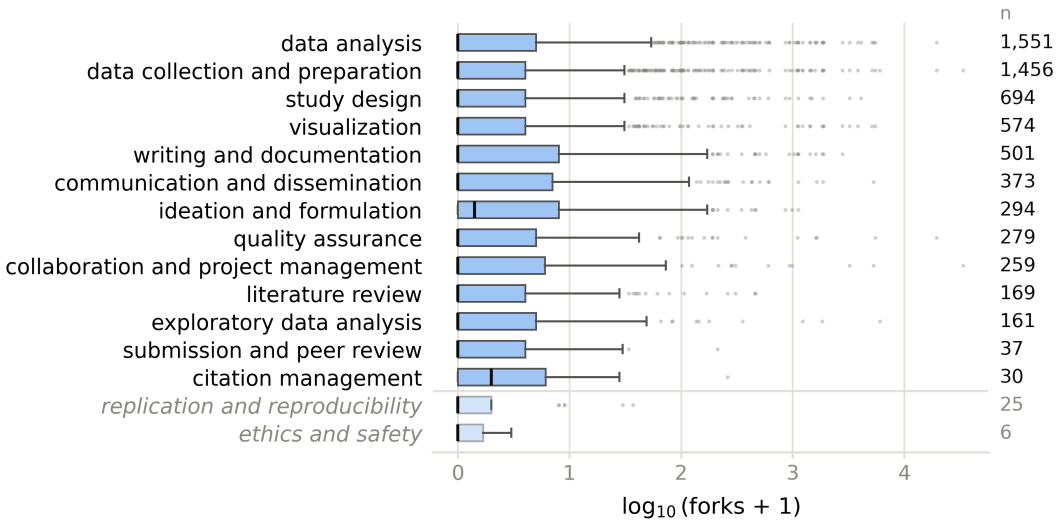


Figure 5: Distribution of the 15 research workflow stages across the 2,200 research-relevant repositories. The annotation is multi-label, so a repository appears under every stage it serves. Execution stages dominate; the sparse end of the lifecycle is ethics and safety (6 repositories), replication and reproducibility (25), citation management (30), and submission and peer review (37).



(a) Stars distribution.



(b) Forks distribution.

Figure 6: Distribution of $\log_{10}$(stars + 1) and $\log_{10}$(forks + 1) by research workflow stage, ordered by repository count (*n*, right). Every stage has a median of 7 stars or fewer and most sit at 1–3, so the long tail is universal rather than concentrated in particular stages. The two grayed rows below the divider hold fewer than 30 repositories each (replication and reproducibility, 25; ethics and safety, 6), so their boxes should be read as small samples rather than distributions.

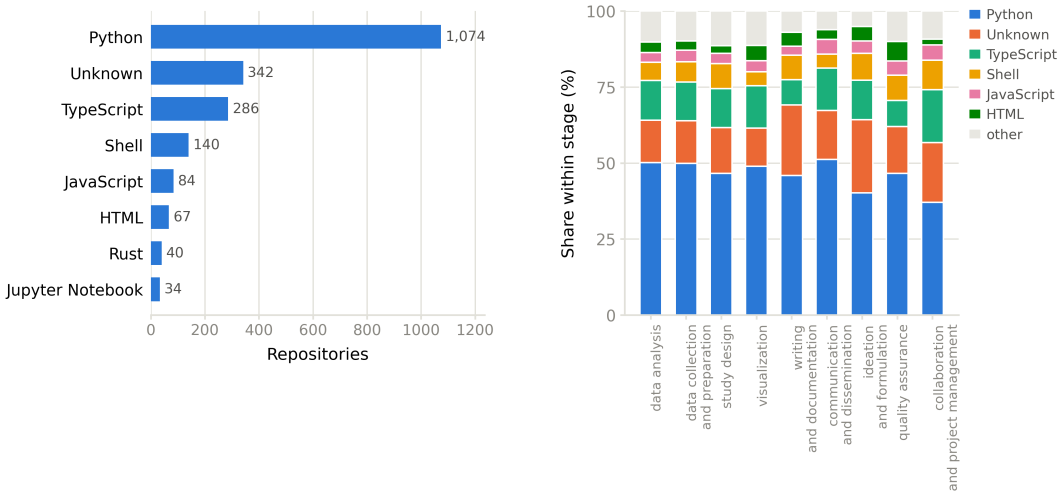


Figure 7: Programming language distribution within research-relevant repositories. Left: the eight most common languages across all research and data-science repositories, showing Python as the dominant language (1,074 repositories), followed by repositories with no language set (342), TypeScript (286), and Shell (140); the remaining 133 repositories are spread across languages below Jupyter Notebook and are not drawn. Right: language breakdown for the nine workflow stages carrying at least 200 repositories, showing that Python is the largest single language in every stage, ranging from 37.1% to 51.2% of repositories. The remaining six stages are omitted because their repository counts do not support a stable per-language breakdown, down to six repositories for ethics and safety. Each column closes with an explicit *other* band so the shares sum to 100%.

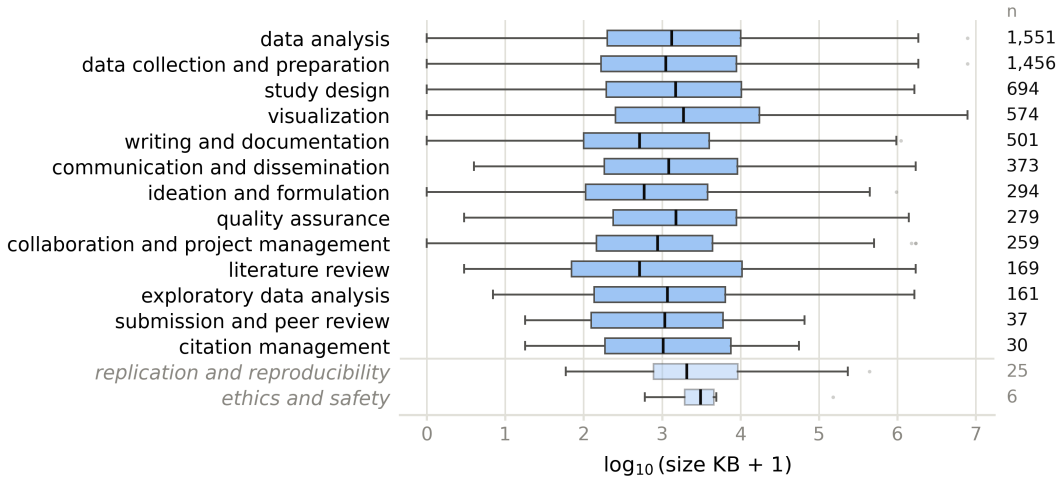


Figure 8: Repository size distribution by workflow stage (log scale), ordered by repository count (n, right); the two grayed rows below the divider hold fewer than 30 repositories each. Box plots show the distribution of $\log_{10}$(size KB + 1) for each stage. Every well-evidenced stage has a median between 0.5 and 2 MB, consistent with repositories containing primarily Markdown skill files and lightweight code. Visualization repositories are the largest of these (median 1,883 KB), ahead of quality assurance and study design, while literature review and writing and documentation sit at the bottom (median 518 KB each).

Table 6: Research workflow stage statistics. *Repos* is the number of repositories tagged with each stage; the annotation is multi-label, so counts sum above 2,200. Every stage has a median of 7 stars or fewer, and most sit at 1–3. These low medians coexist with highly concentrated aggregate stars and forks.

Workflow stage	Repos	Stars (mean)	Stars (med.)	Forks (mean)
ideation and formulation	294	248.6	4	27.5
study design	694	265.1	2	33.8
literature review	169	154.2	1	14.9
data collection and preparation	1,456	568.4	2	81.8
exploratory data analysis	161	529.7	2	66.4
data analysis	1,551	503.2	2	60.8
visualization	574	575.7	2	67.3
writing and documentation	501	400.4	3	45.0
submission and peer review	37	87.4	1	9.9
replication and reproducibility	25	28.0	1	3.8
quality assurance	279	749.7	3	115.0
citation management	30	128.4	7	12.1
collaboration and project management	259	1,005.3	3	183.6
communication and dissemination	373	395.9	3	46.7
ethics and safety	6	1.7	0	0.5

Table 6 provides detailed statistics for each research workflow stage, including repository count, mean and median stars, and mean forks. These statistics complement the box plots of Figure 6 with exact values for cross-stage comparison.

C Pilot Study: Skill-File-Level Triangulation

The large-scale analysis operates on repositories and their README content. To check its conclusions at finer granularity, we conducted a pilot study at the level of individual skill files. We manually curated 7 publicly available repositories[2] that vary in scope, target discipline, and architecture, and identified 389 `SKILL.md` files within them; body lengths range from 462 to 59,009 characters (median 8,871), indicating substantial variation in specification depth. The seven were taken from a list compiled before this study rather than sampled from the crawled corpus, so they are a purposive selection whose spread in popularity (1 to 16,400 stars, Table 8) reflects that curation rather than the ecosystem. Using an LLM-as-judge protocol (Zheng et al., 2023), we scored each skill on its full content (no truncation) along five dimensions: clarity, specificity, research grounding, error prevention, and composability. Clarity (4.95) and specificity (4.91) are consistently high, whereas research grounding (3.59) and error prevention (3.83) show the largest variance and are the dimensions along which quality differs.

The pilot provides a descriptive comparison at a different unit of analysis. Its bottom-up taxonomy is dominated by domain-specific scientific tools: Genomic and Bioinformatics Analysis is the largest type (17.7%), while Research Methodology and Ideation accounts for 4.6%. This resembles the execution-heavy repository-level distribution, although the taxonomies are not directly equivalent. Research grounding and error prevention vary more than clarity and specificity. The purposive selection and shared reliance on LLM judgments mean that this comparison does not rule out sampling or classification artifacts or independently validate the population-level pattern.

Table 7 reports the overall quality-score distribution and Figure 9 the per-type distribution. This appendix reports the curated repositories (Table 8), the bottom-up skill-type taxonomy with per-type scores (Table 9), and per-repository scores (Table 10), supporting the quality analysis of Section 5.4.

[2]The seven repositories, in the order of Table 8, are claude-scientific-skills (`https://github.com/K-Dense-AI/claude-scientific-skills`), knowledge-work-plugins (`https://github.com/anthropics/knowledge-work-plugins`), claude-scholar by Galaxy-Dawn (`https://github.com/Galaxy-Dawn/claude-scholar`), claude-academic-setup (`https://github.com/letitbk/claude-academic-setup`), cc-research-project-template (`https://github.com/mr-devs/cc-research-project-template`), claude-scholar by Y.-Y. Ahn (`https://github.com/yy/claude-scholar`), and Research-Paper-Writing-Skills (`https://github.com/Master-cai/Research-Paper-Writing-Skills`).

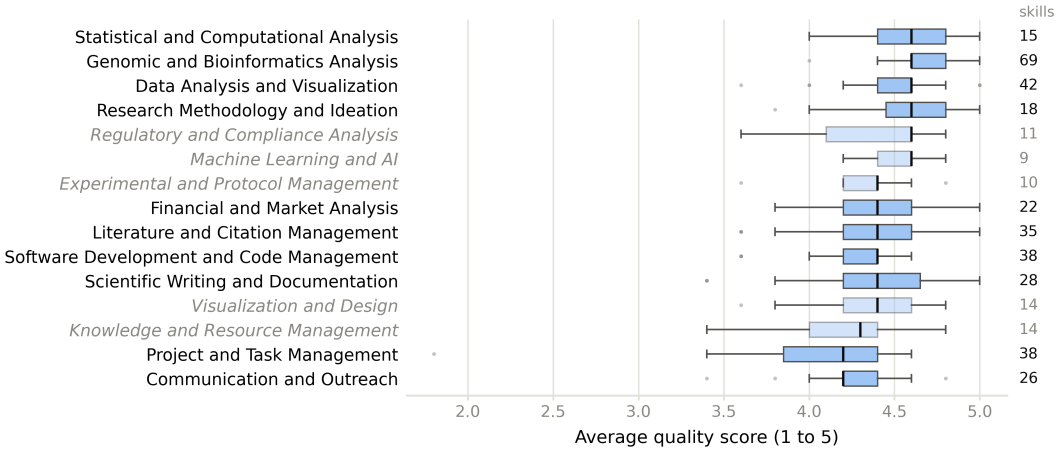


Figure 9: Distribution of the unweighted mean of five GPT-4o specification scores (1–5) for 389 files from seven purposively selected repositories. The 15 bottom-up skill types differ from the workflow-stage taxonomy and are ordered by median; counts appear at right, with groups below 15 files grayed. The composite is not a validated quality index. Repository clustering and file length may influence type differences; scores do not measure executed performance.

Table 7: GPT-4o specification-quality scores for 389 skill files from seven purposively selected repositories (1–5 scale). Standard deviations describe variation across files, not population uncertainty. The final row reports the percentage receiving the maximum score of 5. These scores do not measure executed performance.

Statistic	Clarity	Specificity	Research grounding	Error prevention	Composability
Mean	4.95	4.91	3.59	3.83	4.69
Std	0.22	0.32	0.78	0.71	0.48
Min	3.00	2.00	1.00	1.00	2.00
% at ceiling	95.6	92.0	13.1	13.1	69.2

Table 8: Manually curated pilot repositories, ordered by number of skill files. Star counts were recorded at the time of the pilot collection (counts of 1,000 or more as GitHub displays them, rounded to the nearest hundred), which predates the April 4, 2026 snapshot used elsewhere in the paper, so they differ from the counts in Table 4 for repositories that appear in both; `anthropics/knowledge-work-plugins` does not appear in that crawl.

Repository	Author	Stars	Skills	Focus
claude-scientific-skills	K-Dense AI	16,400	177	Multi-domain scientific
knowledge-work-plugins	Anthropic	10,500	123	Role-based plugins
claude-scholar	Galaxy-Dawn	2,400	47	Academic lifecycle
claude-academic-setup	letitbk	18	22	Social science
cc-research-project-template	DeVerna, M.	1	11	Project structure
claude-scholar	Ahn, Y.-Y.	4	8	Composable tools
Research-Paper-Writing-Skills	Master-cai	902	1	Paper writing
Total		*30,225*	*389*	

Table 9: Bottom-up empirical skill type distribution in the pilot, with average quality scores across all five dimensions.

Skill type	Count	Avg score
Genomic and Bioinformatics Analysis	69	4.64
Data Analysis and Visualization	42	4.49
Software Development and Code Management	38	4.26
Project and Task Management	38	4.04
Literature and Citation Management	35	4.38
Scientific Writing and Documentation	28	4.36
Communication and Outreach	26	4.27
Financial and Market Analysis	22	4.40
Research Methodology and Ideation	18	4.57
Statistical and Computational Analysis	15	4.60
Knowledge and Resource Management	14	4.14
Visualization and Design	14	4.37
Regulatory and Compliance Analysis	11	4.35
Experimental and Protocol Management	10	4.32
Machine Learning and AI	9	4.53
Total	*389*	*4.40*

Judge properties and their bearing on interpretation. Two properties of the LLM-as-judge scores bound how they should be read. First, three of the five dimensions are close to their ceiling: 95.6% of skills score the maximum on clarity, 92.0% on specificity, and 69.2% on composability, against only 13.1% on each of research grounding and error prevention. Research grounding and error prevention provide the most variation, while composability also varies; the unweighted composite should not be read as a validated general quality index. Second, body length correlates with average score ($r = 0.54$). Mean body length differs by a factor of 4.9 across repositories, so specification length may confound both repository and type comparisons. This observational association cannot distinguish a judge's preference for verbosity from additional useful content in longer specifications. Appendix C.1 therefore provides a length-matched non-research comparison, with the uncertainty and research-specific framing of the rubric stated explicitly.

C.1 A Matched Non-Research Baseline

High absolute judge scores require a comparison to interpret. We compare the purposive research pilot with non-research files sampled from a star-stratified repository pool and matched on specification length.

Design. We sampled 100 non-research repositories from an eligible frame of 3,384 among the 11,615 filtered candidates, requiring a strong `SKILL.md` signal. Five star bands contributed

Table 10: Average quality scores by pilot repository (Cla. = Clarity, Spe. = Specificity, Res. = Research grounding, Err. = Error prevention, Com. = Composability).

Repository	Cla.	Spe.	Res.	Err.	Com.	Avg
Research-Paper-Writing-Skills	5.00	5.00	4.00	5.00	5.00	4.80
claude-scientific-skills	4.99	4.98	3.97	3.98	4.94	4.57
claude-academic-setup	4.91	4.91	3.23	4.14	4.27	4.29
knowledge-work-plugins	4.97	4.85	3.36	3.57	4.51	4.25
claude-scholar (Galaxy-Dawn)	4.85	4.83	3.17	3.81	4.51	4.23
cc-research-project-template	4.82	4.82	3.09	4.00	4.18	4.18
claude-scholar (Ahn)	4.88	4.88	3.00	3.50	4.50	4.15

Table 11: Pilot research files ($n = 389$, 7 repositories) and length-matched non-research files ($n = 389$, 30 repositories) from a star-stratified sample, scored with the same judge template and settings on different dates; re-scoring the research files against the non-research run's snapshot leaves every interval spanning zero. Intervals are exploratory repository-cluster bootstrap intervals ($B = 5{,}000$). Perm. p uses 20,000 repository-label permutations and assumes exchangeability of these observational groups. All intervals span zero; the results establish neither equivalence nor a research-quality advantage.

Dimension	Research	Non-research	Diff	95% CI	Perm. p
Generic dimensions					
Clarity	4.954	4.866	$+0.087$	$[-0.010, +0.216]$	0.256
Specificity	4.913	4.851	$+0.062$	$[-0.032, +0.210]$	0.499
Composability	4.686	4.447	$+0.239$	$[-0.104, +0.523]$	0.411
Subset	4.851	4.722	$+0.129$	$[-0.038, +0.280]$	0.396
Research-framed dimensions					
Research grounding	3.594	3.231	$+0.362$	$[-0.121, +0.720]$	0.318
Error prevention	3.830	3.987	-0.157	$[-0.387, +0.095]$	0.360
Subset	3.712	3.609	$+0.103$	$[-0.185, +0.367]$	0.728

8/20/24/24/24 repositories, from the band of 10,000 stars and above down to the band below 10. These quotas span popularity levels but do not reproduce the pilot's star distribution. File-level sampling matched the pilot's body-length decile counts: 389 non-research skills against 389 research skills, with median lengths of 8,944 and 8,871 characters. This addresses the observed length–score association without establishing balance on other properties. The judge uses the same prompt template and settings as the pilot (`gpt-4o`, temperature 0.2, 300 maximum output tokens, no structured-output constraint, full frontmatter-stripped body). Research grounding and error prevention use research-framed rubric anchors, so we report them separately from the generic clarity, specificity, and composability dimensions. Of the 100 repositories, 98 remained reachable, 96 yielded a collected file, and 94 yielded a `SKILL.md`. A test/fixture-path rule excluded 22 files: 19 of 0–112 characters and three of 544, 1,205, and 5,022 characters. No pilot file matched that rule. Research and non-research scoring occurred on March 27 and August 7, 2026, respectively, through the rolling `gpt-4o` alias; the August run records `gpt-4o-2024-08-06` and the March run no snapshot. To test whether the judge changed between the two, we re-scored the 389 research files twice against `gpt-4o-2024-08-06`, sending the March inputs and request with only the model identifier changed (6.61 USD). The March scores agree with the pinned ones as closely as the two pinned passes agree with each other: the integer score is identical on 91.3–99.5% of files per dimension, against 92.3–99.5% between passes, no dimension's March mean differs from the average of the two pinned passes by more than 0.015, and the composite differs by -0.005 (95% paired bootstrap CI $[-0.014, +0.004]$). Repeating the comparison below with either pinned pass leaves every interval spanning zero, with a generic-dimension difference of $+0.139$ or $+0.135$ against $+0.129$, so the difference in scoring dates does not change these results. The 389 matched non-research files come from 30 repositories, with two supplying 336 files.

Results. Table 11 reports descriptive differences and exploratory uncertainty estimates. The generic-dimension difference is $+0.129$ (95% cluster-bootstrap CI $[-0.038, +0.280]$). Research grounding

has the largest positive difference ($+0.362$), but its interval also spans zero ($[-0.121, +0.720]$). These results establish neither equivalence on generic quality nor an advantage in research grounding. Ceiling effects further limit resolution on clarity and specificity.

Robustness and resolution. This is an observational comparison with seven purposively selected research repositories, not randomized treatment assignment. We report a pairs cluster bootstrap, an exploratory repository-label permutation test (20,000 Monte Carlo permutations retaining the 7–30 split), and Welch tests on repository means. The permutation test requires exchangeability of repository labels, which the selection procedure does not establish; it is not design-based randomization inference. Bootstrap and permutation statistics weight files equally, whereas Welch tests weight repositories equally and target a different contrast. No test yields $p < 0.05$. With effective sample sizes of 12.1 and 13.5 after the design-effect correction, the comparison has little power to detect differences of the size at issue, so these non-rejections carry no evidence of equivalence and we do not interpret the individual p-values. The error-prevention difference changes from -0.157 with file weighting to $+0.084$ with repository weighting, illustrating sensitivity to the target of comparison.

Clustering limits both groups. For a model-based diagnostic on the five-dimension composite, we estimate a one-way random-effects ICC using the unequal-size ANOVA correction $m_0 = (N - \sum_i n_i^2/N)/(k-1)$. For the unweighted file mean, a common marginal variance and exchangeable within-repository correlation imply

$$\mathrm{DEFF} = 1 + \left(\frac{\sum_i n_i^2}{N} - 1\right)\mathrm{ICC},$$
$$n_{\mathrm{eff}} = N/\mathrm{DEFF}.$$

The ANOVA correction m_0 is not the cluster-size term in this variance formula. Research ICC 0.247 and size-weighted mean cluster size 126.8 give DEFF 32.1 and $n_{\mathrm{eff}} = 12.1$; non-research ICC 0.181 and size-weighted mean 155.6 give DEFF 28.9 and $n_{\mathrm{eff}} = 13.5$. These composite-score diagnostics are not dimension-specific power calculations or corrections for purposive selection. Broader repository sampling and a stable judge snapshot would be needed for a more informative comparison.

D Annotation Codebook

This appendix gives the full specification of the multidimensional skill annotation schema used in the main text to classify repositories by methodology encoding, AI autonomy level, and fine-grained workflow stage, with results reported in Section 5.

D.1 Multidimensional Skill Annotation Framework

Workflow stage alone leaves finer-grained questions unanswered. For example, two skills tagged as "data analysis" may differ fundamentally in whether they encode methodological judgment or merely wrap a command-line tool, and in whether they assume assistive human oversight or autonomous execution. This subsection gives the full specification of the annotation framework summarized in Section 3 and applied in Section 4 to address such questions. The framework has one repository-level dimension (discipline) and three skill-level dimensions (workflow stage, methodology encoding, AI autonomy level). The codebook defines the skill-level dimensions per skill, but the executed pass applies them once per repository, to its summary (Section 4). The jointly elicited methodology and autonomy labels are associated, mostly because both came from one model call (Section 5.3). The framework is informed by the codifiability–tacit knowledge distinction of Zhang (2026), and the autonomy level dimension is conceptually parallel to the Human Agency Scale of Shao et al. (2025), though measured from the opposite side: our scale codes what a skill *instructs* the AI to do, while HAS codes what workers *want* the AI to do.

D.1.1 Repo-Level Dimension: Discipline

Discipline is coded once per repository rather than per skill. The second annotation pass in Phase 5 receives the same skill summary as the initial pass, together with the repository name and existing stage tags, so it requires no additional data collection. Categories:

Table 12: Broad domain by discipline in the released dataset. Percentages use all repositories within each broad domain, including missing discipline; rounding may prevent totals of exactly 100%. Both fields are model-derived, and *cross-domain* also absorbs off-menu responses.

Discipline	Research ($n = 331$)	Data science ($n = 1{,}869$)
Computer science	183 (55.3%)	1,461 (78.2%)
Cross-domain	74 (22.4%)	176 (9.4%)
Life sciences	34 (10.3%)	64 (3.4%)
Social sciences	23 (6.9%)	44 (2.4%)
Physical sciences	9 (2.7%)	4 (0.2%)
Finance/business	7 (2.1%)	104 (5.6%)
Regulatory/legal	1 (0.3%)	15 (0.8%)
Missing	0 (0.0%)	1 (0.1%)

- *Life sciences*: genomics, bioinformatics, ecology, neuroscience
- *Physical sciences*: physics, chemistry, materials science
- *Social sciences*: economics, political science, psychology, sociology
- *Finance and business*
- *Regulatory and legal*
- *Computer science*: machine-learning and AI research, systems, software-engineering research
- *Cross-domain*: general-purpose research toolkits

The codebook as originally specified allowed interdisciplinary repositories a primary and an optional secondary label (e.g., computational biology → primary: life sciences; secondary: computer science). The executed pass does not implement this: it elicits one discipline per repository, and the released data carries no secondary field. We record the option here because it is the natural extension for a re-run, not as a description of what was produced. Individual skills within a discipline-specific repository that appear generic (e.g., "format LaTeX tables" in a bioinformatics repository) do not require separate discipline coding, since the skill-level dimensions below already distinguish them from domain-specific skills.

Result and its limits. The distribution is heavily concentrated: computer science 1,644 repositories (74.7%), cross-domain 250 (11.4%), finance and business 111 (5.0%), life sciences 98 (4.5%), social sciences 67 (3.0%), regulatory and legal 16 (0.7%), and physical sciences 13 (0.6%). One repository carries no stage tags and is unlabeled.

Table 12 quantifies the overlap: 78.2% of data-science repositories and 55.3% of research repositories are labeled computer science, while cross-domain and life-science labels occur more often in the research stratum. Removing the invalid-summary record changes none of the displayed percentages at one decimal place. The two classifications therefore provide different descriptive detail, but their shared inputs mean that the cross-tabulation is not independent validation of either one.

Four limits bound how far this dimension should be read. First, the concentration is partly definitional: collection targets Claude Code conventions, which can favor software and machine-learning tooling, so *computer science* may be overrepresented and overlaps substantially with the *data science* broad-domain label already reported. Second, discipline inferred from a README-derived summary is a weaker signal than the workflow-stage labels, which describe what a tool does rather than which field it serves.

Third, discipline received no human verification. The pass reported in Section 4 checked methodology encoding, autonomy level, and primary workflow stage; discipline was never put in front of the annotator, so it carries no acceptance rate. The model's own confidence field is available but weak evidence: it reports high confidence on 2,120 of the 2,200 repositories and medium on 79, with no low-confidence label ever emitted and one repository unlabeled, a distribution too saturated to separate reliable from unreliable assignments.

Table 13: The 15 research workflow stages, arranged in six lifecycle groups. *Outputs* covers dissemination artifacts; *communication and dissemination* is under *research-support & infra.* because it covers tools that carry results outward. Counts and percentages show repositories tagged with each stage. Because the labels are multi-label, counts sum above 2,200 and percentages above 100%.

Research workflow stage	Repos (%)
Ideation & design	
ideation and formulation	294 (13.4%)
study design	694 (31.5%)
literature review	169 (7.7%)
Execution	
data collection and preparation	1,456 (66.2%)
exploratory data analysis	161 (7.3%)
data analysis	1,551 (70.5%)
Outputs	
visualization	574 (26.1%)
writing and documentation	501 (22.8%)
Review & QA	
submission and peer review	37 (1.7%)
replication and reproducibility	25 (1.1%)
quality assurance	279 (12.7%)
Research-support & infra.	
citation management	30 (1.4%)
collaboration and project management	259 (11.8%)
communication and dissemination	373 (17.0%)
Governance & ethics	
ethics and safety	6 (0.3%)

Fourth, *cross-domain* is not a pure category. The annotation script coerces any response outside the seven-category list to *cross-domain* and does not retain the raw string, so the 250 repositories (11.4%) it holds mix repositories the model judged to be general-purpose research toolkits with any response that failed to parse or fell off the menu. The two cannot be separated after the fact. Given that the categorization prompt of Phase 3 likewise listed a menu without enforcing it and still drew labels from outside it for 21 repositories (Table 14), we cannot determine how many of the 250 assignments reflect an explicit cross-domain response. This count is not an estimate of independently validated cross-domain prevalence.

We therefore report discipline as a descriptive field of the released dataset rather than as a basis for cross-disciplinary claims, and a re-run that records the raw response and puts the dimension through the verification pass would be needed before it could carry weight.

D.1.2 Skill-Level Dimension 1: Research Workflow Stage

The multi-label workflow-stage scheme used throughout the paper (Table 13), covering 15 stages across the full research lifecycle:

1. Ideation and formulation (hypothesis generation, research question formulation)
2. Study design (experimental design, causal identification, variable mapping)
3. Literature review (search, synthesis, gap analysis, framework selection)
4. Data collection and preparation (scraping, API access, survey design, lab protocols, data cleaning)
5. Exploratory data analysis (descriptive statistics, initial patterns, data quality checks)
6. Data analysis (statistical analysis, computational modeling, estimation)
7. Visualization (figures, plots, interactive displays)
8. Writing and documentation (drafting, editing, formatting manuscripts)
9. Citation management (reference verification, style conversion, bibliography generation)

10. Ethics and safety (IRB compliance, data privacy, AI use disclosure, PII scanning)
11. Submission and peer review (journal selection, reviewer responses, revise-and-resubmit lifecycle)
12. Replication and reproducibility (replication packages, code documentation, clean-run validation)
13. Collaboration and project management (multi-author coordination, CRediT statements, task tracking)
14. Communication and dissemination (presentations, outreach, social media, teaching materials)
15. Quality assurance (claim verification, self-audit, quality gates)

Multi-label coding is recommended because many skills span multiple stages. "Full pipeline" skills that span many stages should be tagged with all applicable stages rather than given a special label.

D.1.3 Skill-Level Dimension 2: Methodology Encoding (Binary)

Does the skill encode methodological judgment and best practices, or is it a procedural tool wrapper?

- *Yes (methodology-encoding):* The skill captures tacit knowledge, research best practices, decision frameworks, or domain-specific methodological guidance. Examples: "apply Bonferroni correction for multiple comparisons," "use PRISMA checklist for systematic reviews," "GATK best practices for variant calling."
- *No (procedural wrapper):* The skill automates a mechanical task without embedding methodological judgment. Examples: "format this table," "run this CLI command," "convert citation styles."

This binary captures a distinction that the workflow stage dimension misses: two skills tagged as "data analysis" can differ fundamentally in whether they encode deep methodology (e.g., multiple testing correction, power analysis) or just wrap a statistical package. Same domain, same stage, but fundamentally different knowledge transfer. A practical heuristic for annotators: *if you removed the skill's instructions and gave the LLM only the task name, would the output miss important methodological guardrails?* If yes, the skill is methodology-encoding; if the output would be largely the same, it is a procedural wrapper.

D.1.4 Skill-Level Dimension 3: AI Autonomy Level

How much independent decision-making does the skill assume for the AI agent? Defined by what decisions the AI makes without human input:

- *Assistive:* Human decides what and how; AI handles mechanics. Example: "format this table."
- *Collaborative:* AI proposes, human disposes. AI generates options or drafts; human selects and validates. Example: "suggest relevant papers."
- *Delegated:* AI executes multi-step workflows with judgment calls; human sets the goal and reviews output. Example: "run this analysis pipeline end-to-end."
- *Autonomous:* AI identifies what to do, executes, and self-validates; human involvement is optional. Example: "monitor dataset for anomalies."

Coding should reflect the *intended* autonomy in the instructions, not demonstrated model capability. Our proposed operational heuristics include counting explicit requests for user feedback and considering what decisions remain with the human. These heuristics were not validated against an external standard, and we are not aware of one for intended agent autonomy; the conceptual framework of Zhang (2026) offers a related perspective rather than a validation of this scale. The Human Agency Scale of Shao et al. (2025) measures workers' desired involvement across five levels; our scale instead codes intended agent autonomy. The two measures are conceptually related but not interchangeable.

Skill-level coding alone misses composition effects: a repository's many individually assistive tool wrappers may be *designed* to chain into delegated pipelines, producing emergent autonomy that no single skill-level label captures. A complementary repository-level annotation pass can therefore distinguish *standalone tools*, where skills are intended for independent invocation, from *composable pipeline stages*, where skills are building blocks for larger orchestrated workflows. This binary can be coded from the repository's README, its inter-skill dependencies, and its documentation of end-to-end use cases. The distinction captures two different design philosophies for complex research workflows: one in which the human orchestrates a pipeline of simple skills, and one in which the AI orchestrates multi-step execution with human review only at the end.

D.1.5 Annotation Protocol and Analytical Payoff

The framework is designed for per-skill annotation with human validation on a stratified subsample. The executed pass differs from this design: GPT-4o-mini coded each repository's summary, and one author checked a stratified subsample with the model's labels visible (Section 4, Appendix A.2). The per-skill, two-annotator protocol below was not executed:

- *Input:* Full `SKILL.md` content (matching the quality evaluation setup in Appendix C).
- *Output format:* One discipline label per repository (the executed pass emits a single label; a secondary label is an option the codebook allows but the run did not implement); multi-label workflow stage; binary methodology encoding; single autonomy level label.
- *Handling ambiguity:* Include a "cannot determine" option per dimension to enable inter-rater reliability measurement.
- *Inter-rater reliability target:* Cohen's kappa of at least 0.7 per dimension with two annotators.
- *Pilot-then-iterate:* Annotate 30 skills together first, discuss disagreements, refine the codebook, then complete the remainder independently.
- *LLM-assisted coding:* LLM coding of full skill files, with human validation on a stratified subsample.

In the main text (Section 4), we applied this framework at scale to all 2,200 research-relevant repositories using GPT-4o-mini, with results reported in Section 5. In the jointly elicited labels, methodology-encoding repositories are predominantly *delegated* (62.5%; Section 5.3). Separate elicitation substantially weakens that association, so it does not establish a property of the underlying repositories.

D.1.6 Design Rationale

Two design choices define this framework. First, discipline is coded once per repository to retain collection-level context and reduce annotation burden. A repository label may miss discipline differences among its files, so it does not make per-skill discipline coding redundant. Second, methodology encoding uses a binary distinction between methodological guidance and procedural wrapping. We selected this dimension to complement workflow-stage labels; we have not established that it is more reliable than a finer-grained scheme or that the sample provides adequate power for every domain-level comparison.

E Extended Research and Labor Context

This appendix extends the Related Work (Section 2) on tacit knowledge, the consequences of AI-assisted research, the labor market, and skill ecosystems. One strand of research locates automation's limit in tacit knowledge. Studying automated paper reproduction, Li et al. (2026a) argue that a key bottleneck is the tacit implementation knowledge papers omit. Zhang (2026) makes a related argument for social science, placing the boundary between human and agent work within each stage, between codifiable execution and tacit judgment, not between stages. In one self-reported physics project, Schwartz (2026) completed a calculation with an agent in two weeks, spending 50–60 hours on oversight. He estimates that the same project would have taken him and a second-year graduate student one to two years, a counterfactual estimate rather than an observed control

condition. One response is to package hundreds of curated medical-research skills for a single field in a domain-specific skill library (AIPOCH, 2026).

Another strand considers consequences of AI-assisted research. Observational work associates AI use with narrowed collective scientific focus (Hao et al., 2026), and a many-analyst study of AI coding agents reports model-dependent differences in analytical choices (Gao & Xiao, 2026). A first-person essay weighs "hiring" AI instead of a graduate student (Rosenfeld, 2026); a researcher survey records concerns about deskilling (Liao et al., 2025). Reporting discusses strain on publishing (Gibney, 2026), while a proposed "100× research institution" explores possible changes in research organization (Hall, 2026).

Generative AI's labor-market impact frames why codified research skills matter. Task-exposure estimates suggest that LLMs could affect much of the workforce (Eloundou et al., 2024). Anthropic's usage report describes geographically uneven diffusion and uncertain projections of faster US convergence than past technologies (Appel et al., 2026). The wage effects of task automation depend on task bundling and on workers' full skill profiles (Freund & Mann, 2025); related commentary also frames AI job changes in terms of bundles (Murgia & Burn-Murdoch, 2026). Experimental productivity effects vary across tasks (Haslberger et al., 2025); observational evidence also associates AI use with larger gains among experienced developers (Daniotti et al., 2026). A study of US data from 2015–2022 estimates different wage effects for augmentation and automation exposure (Marguerit, 2025), while freelance-market evidence finds weaker demand for substitutable skills and novice workers (Teutloff et al., 2025). Massenkoff & McCrory (2026) report tentative evidence of slower job-finding among 22–25-year-olds in exposed occupations, without a detected unemployment effect. Reporting on research jobs raises related concerns about data-analysis and modeling positions (Chen, 2026); it does not establish an academic employment trend. These findings motivate a theoretical verification-bottleneck account: falling execution costs may shift the constraint toward human verification and threaten apprenticeship pathways (Catalini et al., 2026).

Agent skills are a form of reusable agent prompts or instructions. Related work develops prompting techniques and agent architectures. Topics include chain-of-thought prompting (Wei et al., 2022), retrieval-augmented generation (Lewis et al., 2020), multi-agent orchestration (Wu et al., 2024), and tool use (Schick et al., 2023; Shen et al., 2023). These approaches primarily evaluate prompting mechanisms or system architectures. Shareable, version-controlled skill files instead make prompt configurations codified, installable, and community-maintained, much like package managers for agent behavior. Marketplaces and curated collections of skills exemplify the trend (Vercel, 2026).[3]

Recent studies have begun to map the skill ecosystem, but along axes orthogonal to ours. For example, Jiang et al. (2026) measure the prevalence of harmful skills across 98,440 skills in two registries; Liu et al. (2026b) test agents on 263 risky skills gathered from public marketplaces and sorted into seven threat categories; a practitioner report describes 61,776 catalog entries, reporting a GitHub-star Gini coefficient of 0.983 and 54% with zero stars (Zhu, 2026); and Li et al. (2026b) benchmark whether skills improve agent task performance. At the broader tool layer, Stein (2026) characterizes 177,436 Model Context Protocol (MCP) tools, with 5% classified as scientific research and technical analysis. Closer to our work, Destefanis et al. (2026) mine 3,797,117 `SKILL.md` files from 282,200 public GitHub repositories into a dataset that retains file occurrences and groups them by content hash. The dataset stores text, front matter, folder contents, and repository metadata once per distinct file content, and it does not curate by research workflow. File-occurrence and repository counts are different units. Wang et al. (2026) mount the closest curation effort, deploying a six-stage pipeline that distills the skill files of 25,159 GitHub repositories into a curated set of 96,401 and organizing them under a 16-class functional taxonomy scored for utility, robustness, and safety. Its purpose is to serve skills to agents at run time, so its axes are function and quality. More than half of that corpus falls in four of its categories: Dev, Data, Writing, and DevOps-Infra. We instead ask what the research community has codified, which requires a research-workflow axis rather than a functional one.

[3]For example, K-Dense AI's scientific-skills collection (`https://github.com/K-Dense-AI/claude-scientific-skills`) and Galaxy-Dawn's claude-scholar (`https://github.com/Galaxy-Dawn/claude-scholar`).

F Additional Collection and Annotation Details

This appendix expands the collection and annotation procedures summarized in Section 4.

F.1 Phase 1: Repository Discovery

We searched GitHub on April 2–3, 2026, and fetched READMEs on April 4, the snapshot date used throughout. We targeted Claude Code file conventions via the code search API (`SKILL.md`, `.claude/` directories, `CLAUDE.md`, and `.claude/settings.json`), plus repository-level keyword searches (e.g., "claude code skills," "claude-scholar"), which match repository names, descriptions, and topics rather than file contents, restricted to repositories active after Claude Code's launch on February 24, 2025. Because code search caps results at 1,000 per query, we used nine base queries varying in specificity and fork inclusion, adding content filters (e.g., "claude") to broad queries to suppress non-Claude false positives. Broad code queries were adaptively partitioned by file size. Repository searches also required creation after the launch date and retained at most 500 hits per query. The combined routes yielded 23,585 unique repositories after deduplication, of which 23,569 returned metadata; the time filter removed none.

F.2 Phase 2: README-Based Filtering

Many candidates were large software projects (e.g., Electron, PyTorch, Supabase) that contained a `CLAUDE.md` file but were not skills collections. GitHub repository-size metadata summed to 1,068 GiB across the 23,569 candidates, so we fetched their READMEs directly through the GitHub API. Of these, 21,494 had a README and 2,075 did not.

We then devised a weighted keyword-matching score for AI agent skill signals, applied to each repository's README text and its description. Keywords were weighted by specificity: strong (weight 3, e.g., `SKILL.md`, `.claude/skills`, "install skills"), medium (weight 2, e.g., `CLAUDE.md`, `skills.sh`, "agent instructions"), and weak (weight 1, e.g., "claude code," "anthropic"). We retained repositories scoring 1 or higher. This left 11,615 candidates, 50.7% fewer than the initial set, and reduced the aggregate metadata-reported size from 1,068 GiB to 290 GiB; these are not measured download volumes.

F.3 Phase 3: Domain Categorization

For each of the 11,615 filtered repositories, Phase 3 prompted for exactly one of 10 broad domains: *research*, *data science*, *education*, *design*, *industry*, *DevOps*, *web development*, *mobile*, *general coding*, or *other*. The prompt listed the menu but did not enforce it, so 21 repositories carry domain strings outside that list, which we fold into *other* together with the 75 the model labeled *other* (Table 14). All candidates pass to summarization in Phase 4; the 2,200 labeled *research* or *data science* form the research-relevant subset annotated in Phase 5.

The broad-domain assignment was produced by GPT-4o-mini, which read each repository's name, star count, description, language, topics, and a 2,000-character README excerpt (temperature 0.1, at most 200 output tokens) and returned a JSON response that the prompt requested. As a transparency check, we also ran a keyword-matching baseline. For instance, to identify research repositories, we matched keywords such as "experiment," "hypothesis," and "manuscript" against each repository's description, topics, and the first 5,000 characters of its README. The assignments disagree substantially (Appendix G); neither has an accuracy estimate, and the released corpus uses the LLM categorization.

F.4 Phase 4: Summarization

The summarization notebook uses GPT-4o-mini to generate a two-to-four-sentence description of each repository's skills and capabilities from its name, description, language, topics, broad-domain label, and the first 3,000 characters of its README (temperature 0.3, at most 200 output tokens). The notebook is unexecuted, but its selected keys match the archived 2,200-record checkpoint and its request construction is independently corroborated by the executed all-candidate notebook. Across 11,615 candidates, 11,613 carry valid summaries; the two failures were stored as error strings, one

Table 14: Domain distribution of the 11,615 repositories (LLM categorization). The *other* category merges the 75 repositories the model labeled *other* with 21 carrying labels outside the 10 domains listed in the table (`game_dev`: 8, `game-development`: 3, `gaming`: 2, `productivity`: 2, and six singletons). Folding off-menu labels into *other* does not change membership of the subset selected by the *research* and *data science* labels; it does not establish that excluded repositories lack research-relevant content. Percentages are of all 11,615 repositories and may not sum to 100% because of rounding.

Domain	Repositories (%)	
DevOps	3,872	(33.3%)
General coding	2,015	(17.3%)
Data science	1,869	(16.1%)
Web development	1,257	(10.8%)
Industry	1,194	(10.3%)
Education	538	(4.6%)
Research	331	(2.8%)
Mobile	245	(2.1%)
Design	198	(1.7%)
Other	96	(0.8%)

inside the released corpus. The same stored summaries supply both annotation and embedding. We summarize before either branch because READMEs vary in length and contain skill-irrelevant boilerplate, including installation instructions, badges, and licenses. The broad-domain input can also shape which capabilities the summary emphasizes, so downstream differences between domains may reflect both repository content and this conditioning. Appendix A.1 records the provenance of the shared summaries. Appendix A.7 tests omission of the explicit domain field in reconstructed pipelines using GPT-4o-mini, GPT-4o, and GPT-5.

F.5 Phase 5: Multidimensional Annotation

We applied the annotation framework of Section 3 to all 2,200 research-relevant repositories. The initial annotating model read only each repository's *skill summary* (median 544 characters), generated in Section 4. It did not read any `SKILL.md` file. One repository is an exception: its summarization call returned an API error, which was stored in the summary field in place of a description (Section 4), so the labels recorded for it were produced from that error text rather than from any account of its contents and should be read as invalid; it is also the one repository excluded from the 2,199 that carry a summary and an embedding. GPT-4o-mini (temperature 0, parallelized across 50 threads with checkpointing) annotated each summary for methodology encoding, AI autonomy level, and the 15 multi-label research workflow stages in Table 13. The prompt included the codebook's methodology definitions, examples, and heuristic and its autonomy definitions, and listed the 15 stage identifiers without descriptions (Appendix A.9).

Primary-stage and discipline pass. A second pass received the same skill summary, the repository name, and the initial workflow-stage tags to produce two additional repository-level fields used in the analysis. The first, *primary workflow stage*, is a single label chosen from those tags. It determines the colors in the right panel of Figure 2 and Figure 10 and is checked in Phase 7. The model selected a primary stage for the 2,108 repositories with multiple stage tags. For the 91 repositories with a single tag, that tag was taken as the primary stage and the call decided only the discipline. One repository had no stage tag and remained unlabeled. The second is *discipline*, the scientific field the repository serves (Appendix D.1.1). Both labels are elicited in one call per repository, so neither required new data collection, and the prompt is reproduced in Appendix A.9.

G Additional Landscape and Interpretation Details

A keyword-matching baseline using 5,000 README characters agreed with the LLM on only 25.0% of domain assignments. The disagreement is one-sided: the baseline labels 1,383 repositories *research* against the LLM's 331, with 266 in common, because its research keyword list matches generic

README text, and one of its terms, "thesis," also matches inside "hypothesis" and "synthesis." Because the broad-domain label was not put to the human annotator (Section 4), neither arm carries an accuracy estimate; we report the baseline as a transparency check rather than as a validation, and use the LLM assignments, which interpret metadata and a README excerpt in context, throughout. The inputs also differ: the LLM sees a 2,000-character excerpt and additional metadata, whereas the keyword baseline sees 5,000 README characters.

Figure 2 shows the UMAP projection of the 11,613 successfully summarized repositories. Its axes and inter-cluster distances are not quantitatively meaningful, so we interpret only qualitative co-location. The summarization notebook includes the broad-domain field, so domain-colored organization should not be treated as independent validation; historical request and response logs are unavailable. When colored by domain, DevOps, web development, and data science occupy largely distinct regions, and research repositories form a tight band inside the data-science region rather than a region of their own, with a thin scatter reaching across the rest of the map. When colored by primary workflow stage, data collection and preparation co-locates with data analysis. This is a qualitative observation, not a validation of functional separation in the embedding. Their overlap in this projection does not determine whether a classifier can separate their contents.

We report the ecosystem on a single scheme of 15 research workflow stages, arranged in six lifecycle groups, with counts and percentages (Table 13); Table 5 restates the same distribution in rank order and Figure 5 plots it (both in Appendix B). The annotation is multi-label and near-complete: it assigns at least one stage to 2,199 of the 2,200 repositories, yielding 6,409 assignments at a mean of 2.91 per repository.

The landscape is dominated by execution: data analysis covers 70.5% of repositories and data collection and preparation 66.2%, followed by study design (31.5%) and visualization (26.1%). The three stages of the Execution group account for 49.4% of all assignments, and 87.4% of repositories touch at least one of them. The thin end of the lifecycle is thinner still: ethics and safety reaches 6 repositories (0.3%), replication and reproducibility 25 (1.1%), citation management 30 (1.4%), and submission and peer review 37 (1.7%). Percentages are of all 2,200 repositories and do not sum to 100% because a repository is tagged with every stage it serves.

Community engagement follows a pronounced long-tail distribution. Plotting stars against forks (Figure 10) shows strongly correlated engagement proxies, with most repositories near the low-count end. Broken out by stage (Figure 6), the long tail is universal: every one of the 15 stages has a median of 7 stars or fewer, and most sit at 1–3. The concentration is extreme: the 10 most-starred repositories hold 53% of all stars and the 10 most-forked 61% of all forks, the top 1% hold 72% of stars, and 31% had zero stars at the snapshot, a Gini coefficient of 0.97 for both, comparable in magnitude to the 0.983 reported for a broader, differently scoped catalog (Zhu, 2026). Most repositories have low visibility, while a small number receive most stars and forks. This motivates retaining less-visible projects alongside prominent collections; stars and forks do not measure adoption or content coverage.

Discipline, annotated once per repository, is heavily concentrated: 74.7% of repositories are labeled computer science, 11.4% cross-domain, and the named empirical fields are small, led by finance and business (5.0%) and the life sciences (4.5%). Computer science accounts for 78.2% of data-science repositories and 55.3% of research repositories (Table 12). Appendix D.1.1 reports this overlap and the label's limitations.

G.1 Methodology and Autonomous Cases

Broken down by workflow stage (Figure 3), the methodology-encoding rate varies across stage labels. It is highest where the work is evaluative or design-led: among the stages resting on at least 30 repositories, *submission and peer review* leads at 97%, followed by *study design* and *quality assurance* (both 93%). The two stages the figure grays for thin evidence are comparable, *ethics and safety* at 100% ($n = 6$) and *replication and reproducibility* at 92% ($n = 25$). Lower rates occur in support and infrastructure stages: *citation management* (57%), *collaboration and project management* (59%), and *communication and dissemination* (65%). The execution stages sit between the two at 69–75%. The descriptive correlation with stage volume is $\rho = 0.06$ across the 13 stages holding at least 30 repositories. These stage groups overlap, so we do not treat them as independent observations for inference.

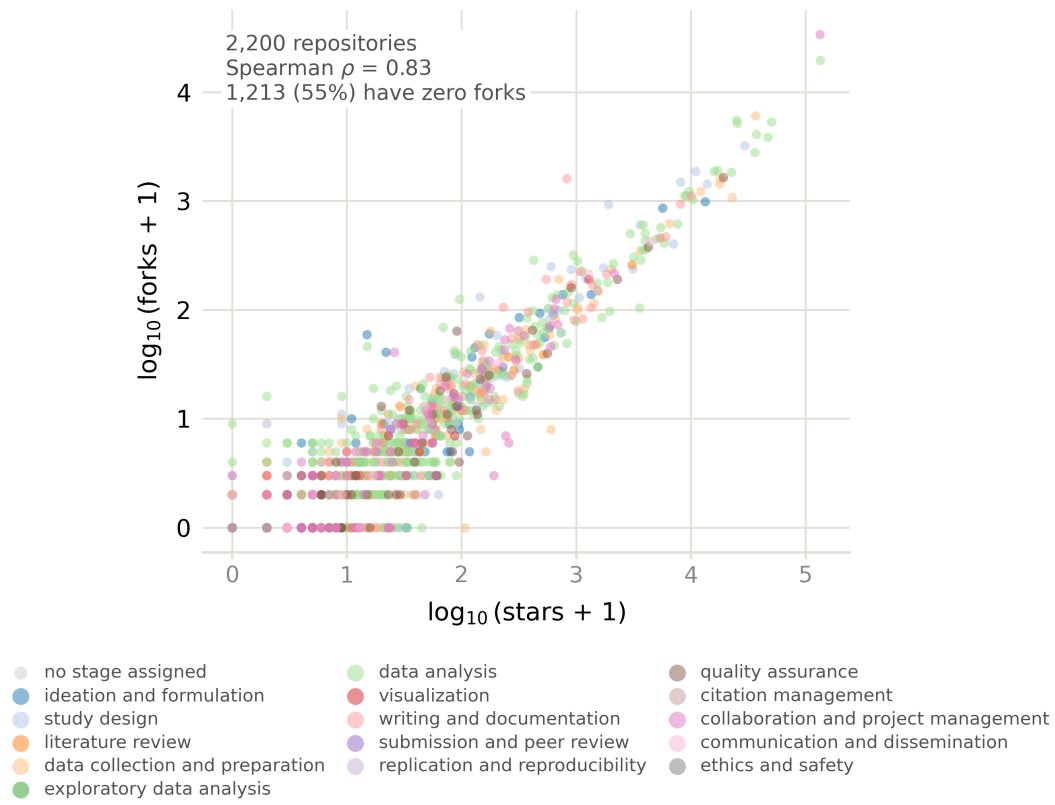


Figure 10: Stars vs. forks for the 2,200 research-relevant repositories ($\log_{10}(x+1)$ axes, retaining zeros), one point per repository. The plot shows two things at once: (i) stars and forks rise together along a single tight band, showing strong correlation between these engagement proxies (Spearman $\rho = 0.83$ over all 2,200 repositories), and no stage sits visibly off that band; and (ii) the band is anchored near the origin, where most repositories sit at low stars and low forks, the long tail of low-visibility projects quantified in the text (Gini 0.97). Points are colored by primary workflow stage; the single repository carrying no stage tag is drawn in gray so that all 2,200 are represented. The stages overlap rather than separating, which is the point: their individual distributions are given in Figure 6.

Only 8 repositories (0.4%) are labeled *autonomous*. An inspection of the eight labeled cases suggests heterogeneous intended uses: three are aspirational end-to-end "idea-to-paper" systems, AutoResearchClaw (10,110 stars), the Sibyl Research System (211 stars), and Sibyl-Codex, and a fourth, Monsets/autoresearch, is a research-specific template running from hypothesis ideation through analysis, while the remaining four are general-purpose self-managing agents only loosely specific to research. These examples qualify the research specificity of the positive cases. Inspecting positives alone does not measure missed autonomous repositories or establish an upper bound on prevalence.

G.2 Specification Quality and Executed Performance

Quality scores also vary descriptively by skill type (Figure 9, Appendix C): genomic/bioinformatics skills average 4.64, statistical-analysis skills 4.60, and project-management skills 4.04. Research grounding has the largest between-type rank statistic ($H = 125.2$), compared with $H = 104.0$ for the composite. These exploratory Kruskal–Wallis statistics treat files as independent; because the files cluster within seven purposively selected repositories, we do not use their nominal p-values as population-level evidence of type differences. The scores assess specification quality, not executed task performance. Ceiling effects, the association with file length, and repository composition limit their interpretation. The observational non-research comparison (Appendix C.1) has wide intervals

spanning zero on every dimension and does not establish either equivalence or a research-grounding advantage.

The distinction from executed performance is consequential: on eight expert-designed literature-review tasks, Hell et al. (2026) report under 6% overlap between expert-selected and AI-selected references for the mid-2025 models they test, with 3% of AI-generated references fabricated and 64% carrying at least one incorrect metadata field, although a single-project check on a 2026 model showed neither failure type. Specification quality and executed performance are therefore distinct measurements, and only the former is in scope here.

G.3 Stratum-Specific Coverage and Intended Oversight

Writing and documentation covers 22.8% of repositories, and communication and dissemination 17.0%, but their distributions differ by stratum. Writing and documentation reaches 38.7% of the 331 research repositories against 20.0% of the 1,869 data-science ones; communication and dissemination reaches 6.3% against 18.8%. Thus the pooled communication figure rests on the data-science majority, while this stage is sparse among research repositories. Likewise, fresh configurations yield lower methodology rates than the released annotations (Section 5.2); the measured extent of methodological guidance depends on the pipeline.

Intended oversight is compatible with the verification-bottleneck account of Catalini et al. (2026), but does not establish actual oversight. Across 263 risky marketplace skills, Liu et al. (2026b) find none of the tested combinations of CLI agent framework and LLM to handle the benchmark reliably; even the configuration with the lowest attack success rate executes unsafe actions in 17.1% of cases. One recurring failure is recognizing a risk but acting before intervening. A human-in-the-loop specification describes intent rather than behavior, and its measured prevalence is prompt-sensitive (Section 5.3).