

COMPLIT: Scientific Literature Search Benchmarks Must Cover Implicit, Cumulative, and Unmet Needs

Dayoon Ko^{1*} Jihyuk Kim² Soyeong Jeong³ Young-Jun Lee⁴ Dahyun Lee²
Juyeon Kim² Gunhee Kim¹ Moontae Lee^{2,5} Kyungjae Lee^{2,6†}

¹Seoul National University ²LG AI Research ³KAIST

⁴University of Minnesota ⁵University of Illinois Chicago ⁶University of Seoul

<https://dayoon-ko.github.io/ComplIt/>

Abstract

How much can we trust an agent to find the paper we need? As researchers, we search the literature from what we have read toward what we have not. We thus describe what we want in our own terms, which a paper may show only implicitly; our requirements grow as we read; and what we seek may lie in an empty region no paper yet covers. We introduce **COMPLIT**, a literature search benchmark spanning all eight arXiv domains, with one diagnostic setting for each of these needs. The strongest agents, Codex (GPT-5.6-Sol) and Claude Code (Opus 4.8), answer 86–87% of STANDARD queries, yet fall to 66–68% when requirements are implicit (IMPLICIT), lose another 6 and 15pp, respectively, as they accumulate (CUMULATIVE), and abstain on at most 53% of the queries no paper satisfies (UNMET). No agent leads in every domain, and all are less accurate on papers from before 2019 or after 2023. Tracing their search, we find that agents often reach the right paper, or the closest one, yet misjudge it. Doing well on COMPLIT is a step toward agents that we can trust to find the paper we need, or to tell us that none exists.

1 Introduction

Researchers increasingly ask agents such as OpenAI Deep Research, Codex, or Claude Code to find papers they need (OpenAI, 2025; Anthropic, 2025). The paper an agent returns shapes what they build on, what they cite, and whether they consider their ideas new (Greenberg, 2009; Pavlovic et al., 2021).

Despite these stakes, most literature search benchmarks (Table 1) evaluate agents from the paper’s side rather than the researcher’s. They build each query from a target paper, so it reuses the paper’s terms, the whole request comes in a single query, and a matching paper always exists. Such queries are thus bound to the target paper: its wording, a single request, and an existing answer.

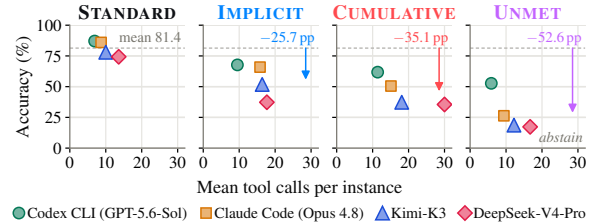


Figure 1: **Frontier agents on COMPLIT.** All four agents score well on STANDARD (mean 81.4%) but drop on every diagnostic setting (arrows: drop in mean). On UNMET, accuracy is the rate of correct abstention.

Researchers, however, search outward, from what they have explored toward what they have not (Kuhlthau, 1991). Because they frame requirements from where they stand, the paper they seek may state them only *implicitly* (Dasigi et al., 2021); as they read on, requirements *accumulate* (Vakkari, 2001); and as they seek what no one has explored, their need may be *unmet* (Asai and Choi, 2021).

In Figure 2, a medical imaging researcher cannot send patient scans outside the hospital. After reading how DeepSeek-R1-Zero learns to reason through reinforcement learning alone (Guo et al., 2025), they look for a method that does the same in-house. A matching paper often describes how it trains without explicitly stating that it does so in-house (*implicit*). Reading that such a paper tests only on X-ray, the researcher also asks for CT and MRI, ruling it out (*cumulative*). If no paper meets them all, the direction may still be open (*unmet*).

We introduce COMPLIT, a benchmark for compositional scientific literature search that evaluates agents in these situations. Alongside a STANDARD reference setting close to existing benchmarks, it has one diagnostic setting per need, each requiring a distinct ability. IMPLICIT requires an agent to *infer* whether a paper meets a requirement from how the paper describes its work. CUMULATIVE requires an agent to *reassess* its earlier answers and move beyond them as requirements accumulate over turns. UNMET requires an agent to *abstain* when no paper satisfies every requirement.

*Work was done during internship at LG AI Research.

†Corresponding author.

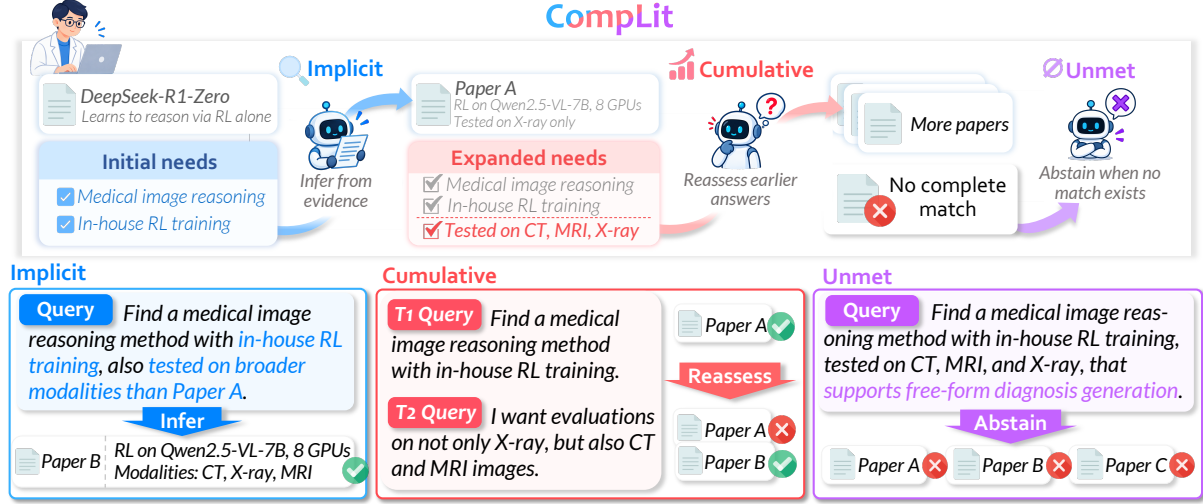


Figure 2: **Motivation and overview of COMPLIT.** **Top:** as a researcher reads retrieved papers, their need proves *implicit*, *cumulative*, and *unmet*: the required property is rarely stated outright, each new requirement rules out earlier candidates, and the full set may match no paper. **Bottom:** our three diagnostic settings test whether (i) an agent *infers* such a property from paper evidence, (ii) *reassesses* earlier answers as requirements accumulate, and (iii) *abstains* when no paper meets the need.

We compose each query from several requirements taken from survey comparison tables, and collect 870 single-turn queries and 220 conversations, all manually verified, across all eight arXiv domains. Beyond final accuracy and its breakdown by domain and publication year, COMPLIT traces agents’ search dynamics over tool calls or turns for each setting, pinpointing where they go wrong.

The strongest agents already reach 86–87% accuracy on STANDARD (Figure 1), where, as in existing benchmarks, most requirements are stated outright, all arrive at once, and a match always exists. Even there, average accuracy ranges from 74% in physics to 94% in finance and economics, and falls from 88% on papers from 2019–23 to 75% on older and 80% on more recent ones (Table 3). Beyond these gaps, each diagnostic setting reveals a different way they fall short.

On IMPLICIT, agents struggle to *infer* whether a paper meets a requirement from indirect evidence. The gold paper is retrieved less often, and even when it is, agents often leave it unread, or read it and still choose another paper (Figure 3). In turn, accuracy drops 19 to 37pp, and even the best agent returns a wrong paper for about a third of queries.

On CUMULATIVE, agents struggle to *reassess* their way to the right paper, with Claude Code (Opus 4.8) and Codex CLI (GPT-5.6-Sol) failing in different ways. Claude Code often leaves a wrong answer but keeps picking new wrong papers, still losing 15pp from the corresponding single-turn queries, while Codex picks a wrong paper far less

often in the first place, yet still loses 6. As a result, the two, which score alike on single queries, end 11pp apart (Figure 4).

On UNMET, agents struggle to *abstain* when no paper qualifies. They open the paper closest to the request within ten tool calls in over 90% of queries, yet still present a paper as a match in 47% to 82% of queries (Figure 5). A researcher may thus mistake an open direction for one already taken.

In summary, our contributions are as follows:

- We introduce COMPLIT to evaluate scientific literature search on researchers’ implicit, cumulative, and unmet needs across eight domains.
- We test thirteen baselines, from corpus retrievers to agents, reporting trends by domain and publication year.
- We also analyze search dynamics to pinpoint where each agent fails, not only in retrieving a paper but also in verifying it.

2 Related Work

Literature Search Benchmarks. Early benchmarks established natural-language retrieval as a setting for scientific literature search. LitSearch (Ajith et al., 2024) and PaSa (He et al., 2025) generate queries automatically from citation contexts, each adding a smaller researcher-written set, and PaperFindingBench and LitQA2-Search (Bragg et al., 2026) draw queries from real user logs and from biology questions. Each of these benchmarks poses a request as a single need rather than a composition of constraints.

Benchmark	Constraint		Task			Domains	Scope	Q
	Type	Level	Implicit	Cumulative	Unmet			
LitSearch (2024)	①	E P	×	×	×	CS	Abs	597
PaSa (2025)	①	P	×	×	×	CS	Abs	1,050
DORIS-MAE (2023)	①	E P	×	×	×	CS	Abs	100
PaperFindingBench (2026)	①③	E P	×	×	×	CS	Full	267
LitQA2-Search (2026)	①	P	×	×	×	BIO	Full	75
PaSaMaster-Bench (2026)	①③	E P	×	×	×	CS, EE, PHY, MATH, BIO, ECO	Full	244
SAGE (2026)	①②③④	E	×	×	×	CS, PHY, BIO, ECO	Full	1,200
AutoResearchBench-Deep (2026)	①③④	P	×	×	✓	CS	Full	600
AutoResearchBench-Wide (2026)	①③	E P	×	×	×	CS	Full	400
COMPLIT (Ours)	①②③④	E P I	✓	✓	✓	CS, EE, PHY, MATH, BIO, ECO, FIN, STAT	Full	870 + 220

Table 1: **Comparison of COMPLIT with scientific literature search benchmarks.** Constraint *Type* is ① content, ② figural, ③ metadata, or ④ comparative, a relation to another paper; *Level* is how the target paper states a constraint, Explicit, Paraphrased or Inferred, and COMPLIT alone spans all three. *Implicit*, *Cumulative* and *Unmet* are three diagnostic tasks. *Domains* follow arXiv’s top-level taxonomy, and *Scope* is the part of a paper needed to answer: an abstract or a full paper. |Q| counts evaluation queries; COMPLIT has 870 single-turn queries and 220 conversations.

Multi-Constraint Queries. A complementary line structures complex needs into multiple aspects or conditions. DORIS-MAE (Wang et al., 2023) decomposes scientific queries into hierarchical aspects, and PaSaMaster-Bench (Du et al., 2026) pairs each expert-written intent with a checklist of criteria across six domains. Beyond scholarly search, BrowseComp (Wei et al., 2025) combines hard-to-find factual web clues, while MultiConIR (Lu et al., 2025) scales queries from one to ten conditions. Recent deep-research benchmarks have brought this structure to the scientific literature. SAGE (Hu et al., 2026) composes queries from paper artifacts and citation overlaps, while AutoResearchBench (Xiong et al., 2026) derives Deep and Wide queries from target papers.

By contrast, we start from the researcher’s side, where a request requires an agent to *infer* evidence a paper leaves implicit, to *reassess* candidates as the requirements accumulate, and to *abstain* when no paper meets the need.

Implicit, Cumulative, and Unmet. Each of our diagnostic settings has a counterpart outside scholarly search. Retrieval and reading comprehension benchmarks make relevance depend on facts a document states only implicitly (Taghavi et al., 2025; Tchuindjo et al., 2026). Conversational benchmarks reveal the need over turns and show that models anchor on early assumptions (Ali et al., 2026; Laban et al., 2026). Unanswerable questions test whether models abstain (Rajpurkar et al., 2018; Kirichenko et al., 2026). Scholarly search demands all three as a researcher explores what they have not: properties are rarely stated outright, requirements grow with each paper read, and their

conjunction may match no paper. Yet no scholarly benchmark in Table 1 tests them. While Xiong et al. (2026) covers abstention, only 60 of its queries do so, all in computer science. COMPLIT tests each ability in a dedicated setting, and its analysis of search dynamics shows not only whether an agent fails but where in the search it does (§4.4).

3 The ComplLit Benchmark

We propose COMPLIT, a benchmark for compositional scientific literature search. We define the task (§3.1), derive the constraints of each query (§3.2), build a STANDARD and three diagnostic settings (§3.3), and provide a corpus (§3.4).

3.1 Task Definition

A query q expresses a research intent and a set of constraints $\mathcal{C} = \{c_1, \dots, c_k\}$ that a paper the researcher has not yet found must satisfy jointly. A system searches a corpus $\mathcal{D}$ of papers through a fixed index or the open web.

A paper $p \in \mathcal{D}$ satisfies c_i , written $p \models c_i$, if its full paper, with figures, bibliographic record, or comparison with another paper supports c_i , and $p \models \mathcal{C}$ if it satisfies every constraint. The gold answer is the target paper p^* with $p^* \models \mathcal{C}$, or $\emptyset$ if no paper in $\mathcal{D}$ satisfies $\mathcal{C}$. A system returns a single paper or abstains; an answer is correct if it is p^* , or an abstention when the gold is $\emptyset$, and we report accuracy on each set.

3.2 Query Generation Pipeline

We use Qwen3.5-1.22B-A10B (Qwen Team, 2026) as the LLM and the VLM in every step of constraint derivation and query generation, with prompts in Appendix C.1.

Constraint sources. We start with survey comparison tables that record each surveyed paper’s attributes. The survey authors consider distinguishing in their field, so their rows share a scope yet differ on those attributes. Each constraint c_i is a proposition about the target paper with a source $s_i \in \{\text{content, figural, metadata, comparative}\}$.

From the table, *content* constraints are the cell values of the paper’s row, restated as propositions, and *comparative* constraints contrast the paper with another paper in the same table (e.g., broader imaging modalities). From the paper itself, *figural* constraints are claims a vision-language model (VLM) reads from its figures, either quantitative (an ordering, a peak value) or qualitative (an architecture pattern, a trend), and *metadata* constraints are the year, categories, and authors in its arXiv record.

These sources test full-paper understanding, figure interpretation, and cross-paper comparison.

Constraint verification. To ensure factuality and high quality, we verify every constraint exhaustively against the target paper. When $s_i \in \{\text{content, comparative}\}$ we do not know which part of the paper supports the constraint, so a *grounder* LLM reads the paper and records the supporting passage, and a *verifier* scores whether that passage appears in the paper and supports the constraint. When $s_i \in \{\text{figural, metadata}\}$, the evidence is already known: a VLM *verifier* scores each figural claim against its figure and caption, and metadata constraints are checked against the arXiv record.

If the verifier scores a constraint below an empirical threshold of 0.9, we reground or regenerate it and score it again for up to 10 rounds; any constraint that never passes is discarded. Every constraint that passes is then rechecked by the authors, so each clears both automatic and manual reviews.

Evidence levels. We label every content, figural, and comparative constraint with an evidence level $\ell_i \in \{\text{explicit, paraphrased, inferred}\}$, which records how a system must ground it in the paper. We judge it against the paper for a content constraint, and against the figure for a figural one.

A constraint is *explicit* when the system can read the fact as it stands, and *paraphrased* when the system must match meaning in different words, or when the claim re-describes what the figure directly depicts. It is *inferred* when the system must derive the fact, because the text only implies it (e.g., training in-house), or because it follows only from comparing several visual elements (e.g., a ranking

or a trend).

The *inferred* level also covers every comparative constraint, since establishing one requires reading a second paper. These labels are the basis of the IMPLICIT setting (§3.3).

Query writing. We select constraints to fit a research intent, so that a query carries only the properties a researcher would need. An intent is generated from one of 16 search purposes (Haddad et al., 2026), such as methodological guidance or comparative analysis. For each paper, we select up to three intents it can serve and verify each with the same LLM. The LLM then chooses the constraints that serve each intent and writes a request around it. We repeat generation and verification to ensure quality, and record the constraints it uses as $\mathcal{C}$.

Human verification. The authors then check every query three ways: each constraint’s recorded evidence must be in the paper and support the constraint; the assigned intent must fit the paper and claim nothing the paper does not support; and the query must identify the target.

Uniqueness check. We then check automatically that every gold paper is the only known match. For each query, we retrieve the top-50 papers with both BM25 (Robertson and Zaragoza, 2009) and Qwen3-Embedding-8B (Zhang et al., 2025), and the LLM judges whether any of the 100 papers satisfies all constraints. We remove a query if any paper other than the target does, or if any paper at all does for an UNMET query. This removed 5.5%, 4.3%, and 1.0% of queries in STANDARD, IMPLICIT, and UNMET, respectively.

3.3 Diagnostic Settings

The settings follow how a literature search unfolds. Alongside STANDARD, each setting evaluates one challenge.

STANDARD. This setting, close to existing benchmarks, serves as a reference for the diagnostic settings. Queries state constraints at all evidence levels, given at once, and every query has a target. This set supports analysis across domains, publication years, and numbers of constraints. It contains 350 queries over 251 target papers.

IMPLICIT. This setting tests whether a system can *infer* that a paper meets requirements it states only implicitly. Every constraint must be inferred: $\ell_i = \text{inferred}$ for every $c_i \in \mathcal{C}$, so queries carry no metadata constraints. No constraint can then be resolved by lexical or semantic matching. Rather,

each must be reasoned out from what the paper reports or from how it compares to a named paper. It contains 220 queries over 179 papers.

CUMULATIVE. This setting tests whether a system *reassesses* its earlier answers as requirements accumulate over turns. The need is revealed over multiple turns: following (Laban et al., 2026), we rebuild each IMPLICIT query into a conversation of 3 to 5 turns. An offline scheduler (gpt-5.6-sol (OpenAI, 2026a)) partitions $\mathcal{C}$ into disjoint subsets $\mathcal{C}^{(1)}, \dots, \mathcal{C}^{(T)}$, one per turn with one to three constraints each, and writes the user utterance for each turn in advance.

At evaluation time, a gpt-5.4-mini (OpenAI, 2026b) user simulator at temperature 0 delivers each utterance, so every system faces the same conversation with the same constraints at each turn (Laban et al., 2026). The system may answer at every turn and is scored on its final answer, once all of $\mathcal{C}$ has been revealed, with per-turn answers kept for analysis. This set contains 220 conversations.

UNMET. This setting tests whether a system *abstains* when no paper meets every requirement. We construct queries as in STANDARD and retain those answered correctly by all three reference systems, chosen for their low cost: Harness-1 (Jiang et al., 2026), DeepSeek-V4-Pro (DeepSeek-AI, 2026), and GLM-5.2 (GLM-5-Team, 2026). We then add a false constraint $\tilde{c}$ by perturbing a verified constraint of the target that is not in $\mathcal{C}$, such as substituting another paper’s survey table value or reversing a figure’s trend, so that no paper in $\mathcal{D}$ satisfies $\mathcal{C} \cup \{\tilde{c}\}$ and the gold is $\emptyset$. Since every target is thus easy to find, this setting separates abstention from retrieval. It contains 300 queries.

3.4 Corpus

For index-based search, we provide a corpus of 2.95M full-text arXiv papers from the S2ORC v2 snapshot (Lo et al., 2020), with survey papers removed so that corpus-based systems cannot rely on them to find the target, while keeping every target paper. Open-web systems are not restricted, but rarely open the source survey (Appendix D.2).

4 Experiments

We compare thirteen baselines on the four settings (§4.2), then analyze accuracy by domain and publication year on STANDARD (§4.3) and search dynamics on IMPLICIT, CUMULATIVE, and UNMET (§4.4).

4.1 Experimental Setup

Evaluation sets and corpus. We use the four settings in §3.3: STANDARD, IMPLICIT, CUMULATIVE, and UNMET, with 350, 220, 220, and 300 instances. Corpus-based baselines search the 2.95M-paper collection described in §3.4.

Baselines. We evaluate thirteen baselines in three groups by how they acquire candidate papers. (i) **Corpus retrieval:** BM25 (Robertson and Zaragoza, 2009), SPECTER2 (Singh et al., 2023), ReasonIR (Shao et al., 2025), and Qwen3-Embed-8B (Zhang et al., 2025) alone and with Qwen3-Reranker-8B. (ii) **Corpus search agents:** DCI-Agent-Lite (Li et al., 2026), which explores the raw corpus via shell, and Harness-1 (Jiang et al., 2026), which searches it with a state-externalizing harness. (iii) **Web search agents:** DR Tulu (Shao et al., 2026), PaSa (He et al., 2025), DeepSeek-V4-Pro (DeepSeek-AI, 2026), Kimi-K3 (Kimi Team, 2026), Claude Code (Anthropic, 2025) with Opus 4.8, and Codex CLI (OpenAI, 2026a) with GPT-5.6-Sol, which search the open web; we shorten the last four to DeepSeek, Kimi, Claude, and Codex, and refer to them as the frontier agents.

Corpus retrieval always returns a paper and cannot abstain, so we exclude it from UNMET. CUMULATIVE requires holding a conversation across turns, which only DeepSeek, Kimi, Claude, and Codex support, so we run it on these four.

Tools in web agents. DR Tulu and PaSa run with their released pipelines. Kimi and DeepSeek run in `inspect_evals` with three tools: `arxiv_search` with arXiv API, `web_search` through Serper, and `read_paper`, which returns a paper’s full text with figures, given a message limit of 30. Claude and Codex run in their own web search harnesses, plus `arxiv_search` and `read_paper` as MCP tools.

Metrics. Each baseline either returns a paper or abstains, since the task is to find one paper that satisfies all constraints jointly (§3.1). Corpus retrieval answers with its top-ranked paper (i.e., Recall@1), so it is scored comparably to agents. Agents end each run with a JSON object holding a title and a rationale, and an empty title indicates abstention.

We report accuracy, where a prediction is correct if it matches the gold paper’s title, or if the baseline abstains on an UNMET query. On CUMULATIVE we score the answer at the final turn. Kimi and DeepSeek runs that do not commit within the message limit receive a forced final answer, counted toward accuracy and marked separately in Figure 3.

Baseline	STD.	IMP.	CUM.	UNM.
<i>Corpus Retrieval</i>				
SPECTER2	4.6	0.1	–	–
BM25	28.9	3.2	–	–
ReasonIR	15.4	5.5	–	–
Qwen3-Embed-8B	49.1	18.6	–	–
+ Qwen3-Rerank-8B	66.0	25.9	–	–
<i>Corpus Search Agents</i>				
DCI-Agent-Lite [†]	54.3	24.1	–	5.9
Harness-1	69.7	42.3	–	9.0
<i>Web Search Agents</i>				
DR Tulu	23.7	7.3	–	14.3
PaSa	42.6	15.9	–	8.7
DeepSeek-V4-Pro	74.3	37.3	35.5	17.3
Kimi-K3	78.0	51.8	37.3	18.7
Claude Code (Opus 4.8)	86.0	65.9	50.5	26.3
Codex CLI (GPT-5.6-Sol)	87.1	67.7	61.8	52.7

Table 2: **Main results on COMPLIT.** Accuracy (%) on STANDARD, IMPLICIT, CUMULATIVE, and UNMET (abbreviated as shown). On CUMULATIVE, the final-turn answer is scored; on UNMET, accuracy is the share of queries on which the baseline abstains. – marks settings a baseline does not support (§4.1). [†]With GPT-5.4 nano. The best score in each group is in **bold**.

4.2 Main Results

On STANDARD, the four frontier agents score high, while the rest vary widely within each group and no group is consistently ahead. The frontier agents score 74.3% to 87.1%, with Codex and Claude one point apart. The rest spread widely: corpus retrieval ranges from 4.6% to 66.0%. Interestingly, the reranked retriever, at 66.0%, sits between the two corpus search agents and is even 23pp above PaSa. This set shows which baselines are effective within each group, but not how they can fail when constraints are inferred, cumulative, or unsatisfiable.

On IMPLICIT, all baselines degrade, and even the strongest by about 20pp. Codex and Claude degrade by 19 and 20pp, Kimi by 26, and DeepSeek by 37, showing that the weaker agents degrade more. The reranked retriever degrades the most, by 40pp, whereas the corpus search agents, which browse the same index, degrade by 27–30. This suggests that agents’ reasoning over papers helps establish properties that must be inferred.

On CUMULATIVE, Codex and Claude, which score alike on a single query, end 11pp apart. All four agents degrade relative to IMPLICIT, by 2 to 15pp. In particular, the gap between Codex and Claude widens from 2pp to 11 as constraints are imposed over turns. This suggests that a single-query

	Codex	Claude	Kimi	DeepSeek
<i>Domain</i>				
Physics	87.3	79.4	68.3	60.3
EESS	88.3	83.3	80.0	76.7
Statistics	84.9	90.6	67.9	66.0
Computer science	80.4	84.3	84.3	70.6
Biology	88.4	90.7	81.4	83.7
Mathematics	87.5	82.5	77.5	80.0
Finance & economics	95.0	95.0	92.5	92.5
<i>Publication year</i>				
≤2015	79.5	78.2	65.4	71.8
2016–18	87.3	84.1	73.0	63.5
2019–21	92.6	96.3	87.7	81.5
2022–23	93.8	89.2	81.5	80.0
≥2024	82.5	81.0	82.5	73.0

Table 3: **Accuracy by domain and publication year of the target.** Results are for the four frontier agents on STANDARD. Darker cells mark higher accuracy.

evaluation can obscure the gap between agents in handling cumulative requirements.

On UNMET, most baselines return a paper, and even the strongest fails to abstain on almost half of the queries. Codex abstains correctly on 52.7% of queries and Claude on 26.3%, while every other baseline stays below 20%. Among them, Harness-1, DR Tulu, and PaSa, which are trained on answerable queries, abstain on 9.0%, 14.3%, and 8.7%. This suggests that verifying every constraint and abstaining if even one fails must be trained explicitly.

4.3 Accuracy by Domain and Year

Covering all eight arXiv domains and papers through 2026, COMPLIT can show researchers in each field which agent serves them well and where it may fall short. We break down the STANDARD accuracy of the four frontier agents by domain and publication year (Table 3).

No agent leads across all domains, and each agent struggles with different domains. Codex leads in physics, EESS, and mathematics, and Claude in statistics and biology. Codex scores 85% to 88% in most domains but 80.4% in computer science, below Claude and Kimi. Claude scores above 90% in statistics and biology but 79.4% in physics. Kimi and DeepSeek are weakest in physics and statistics, and DeepSeek is also the lowest of the four in EESS and computer science.

All four are more accurate on papers in 2019–23 than on older or more recent ones. Relative to their best period, the agents score 14 to 22pp lower on papers from 2018 or earlier: Codex, Claude, and Kimi are weakest on papers from 2015 or earlier,

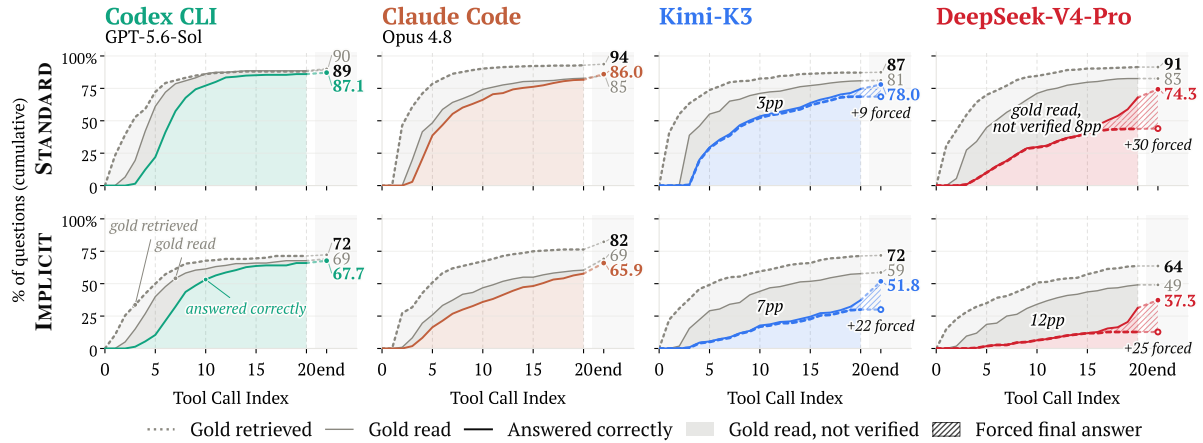


Figure 3: **Search dynamics on STANDARD and IMPLICIT.** By tool call, the share of queries in which the gold paper has been retrieved (dotted), read (solid gray), and answered correctly (colored). The *end* column shows final values. Hatching marks answers that Kimi and DeepSeek give only when forced at the end of the budget.

and DeepSeek on those from 2016 to 2018. On papers from 2024 onward, Codex, Claude, and DeepSeek score 8 to 15pp below their best, and Kimi 5pp below. This shows that agents are less reliable on older and very recent papers, which may be harder to retrieve or verify.

4.4 Analysis of Search Dynamics

4.4.1 STANDARD vs. IMPLICIT

Figure 3 tracks, over tool calls, the share of queries in which the gold paper has appeared in search results (*retrieved*),¹ has been opened (*read*), and has been given as the answer (*answered correctly*), on STANDARD and IMPLICIT.

On IMPLICIT, agents fail more often at every step, across retrieving, reading, and answering with the gold. In Figure 3, all three curves shift down and to the right from STANDARD to IMPLICIT: at the same number of tool calls, the gold paper is retrieved, read, or answered in fewer queries. The final share of queries with the gold paper *retrieved* falls by 12 to 27pp, the share with it *read* by 16 to 34pp, and accuracy by 19 to 37pp.

The gaps between the steps also widen, both from *retrieved* to *read* and from *read* to *answered correctly*. This suggests that when every constraint must be inferred, agents find it harder both to decide which paper to read and to verify every constraint in a paper they have read.

The weaker the agent, the wider the gaps between retrieving, reading, and answering with the gold paper. On IMPLICIT, the gap between

retrieving and reading the gold paper is 3pp for Codex and 13 to 15pp for the other three agents, and the gap between reading and answering grows from 1 point for Codex to 3, 7, and 12pp for Claude, Kimi, and DeepSeek. Also, Kimi and DeepSeek answer many of the papers they read only when forced, recovering 22 and 25pp of their accuracy. This suggests that agents have room to improve not only in retrieving the gold paper but also in reading and answering with it once retrieved.

4.4.2 Revision on CUMULATIVE

Figure 4 tracks, by user turn, the share of conversations whose current answer is the gold paper (*committed to gold*) or another paper (*on another paper*), the share that has chosen another paper at some turn (*ever chose another*), and their difference (*escaped*).

Across turns, agents differ in how they revise their answers, and wrong answers accumulate for the weaker agents. Two questions separate the agents: whether wrong answers rise faster than correct ones, and if so, whether the agent escapes them. For Codex, correct answers rise faster, leaving few wrong ones to escape. For the other three, wrong answers lead from the second turn. Claude escapes many of them, so the share of conversations currently on a non-gold paper (solid gray) remains near 40%, even though the share that has ever chosen one (dotted) continues to rise. Kimi and DeepSeek escape a few, and their current wrong share rises to 61%. These differences emerge only as requirements arrive over turns, so a single-turn evaluation cannot show whether an agent revises its earlier answers.

¹Codex does not expose its native web search results, so we count only `arxiv_search` and `read_paper` calls.

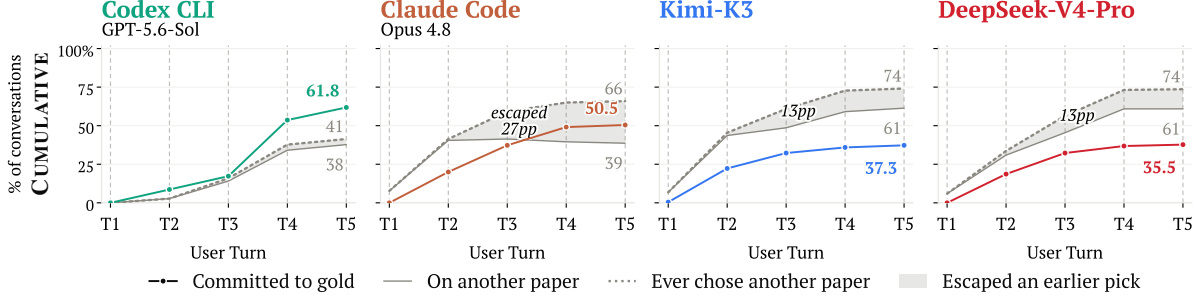


Figure 4: **Revision dynamics on CUMULATIVE.** By user turn, the share of conversations whose current answer is the gold paper (colored) or another paper (solid gray), and the share that has chosen another paper at some turn (dotted). The shaded band shows conversations that chose a wrong paper at an earlier turn but later escaped it.

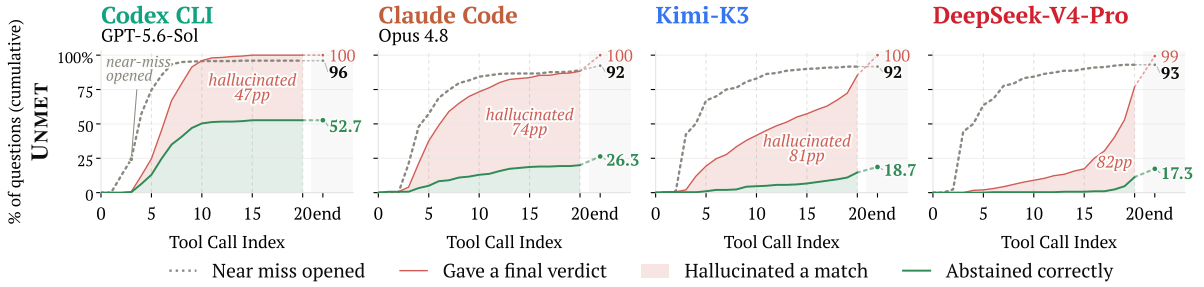


Figure 5: **Abstention dynamics on UNMET.** By tool call, the share of queries in which the agent has opened the near-miss paper (dotted), has given a final verdict (red), and has abstained correctly (green). The shaded area shows verdicts that name a paper although no paper satisfies all constraints.

4.4.3 Abstention on UNMET

Each UNMET query has a *near-miss*, the original target that satisfies every constraint except the injected one. Figure 5 tracks, by tool call, whether the agent has opened the near-miss (*near-miss opened*) and given its final answer (*gave a final verdict*). That answer either names a paper, the near-miss or any other, as satisfying every constraint (*hallucinated a match*), or abstains (*abstained correctly*).

All agents open the near-miss early, but most go on to accept it as a match. All four agents open the near-miss in over 90% of queries within the first ten tool calls. Yet only Codex abstains in about half of the queries, while the other three end with a paper, nearly always the near-miss, in 74% to 82% of queries. Since UNMET keeps only queries whose target the reference baselines already find (§3.3), the results suggest that agents fail at verification: they find the near-miss but do not abstain.

Agents open the near-miss at the same point, but decide at different ones. Codex commits within a few calls of opening it, whether it abstains or not, while Claude and Kimi decide gradually and DeepSeek mostly in its last few calls. The later an agent decides, the less often it abstains, from 52.7% for Codex to 17.3% for DeepSeek. Stronger

agents thus seem to better gauge their confidence in the near-miss (Yoon et al., 2025).

5 Conclusion

We introduced COMPLIT, a benchmark that evaluates agents in the way researchers search: outward from what they have read, with needs that are implicit, cumulative, or unmet. Even the strongest agents lose about 20 points when every constraint is implicit and fall further, each in a different way, as requirements accumulate, while most agents accept a near-miss instead of abstaining. Beyond final accuracy, COMPLIT traces each agent’s search step by step, revealing where these failures arise: agents often reach the right paper, or the closest one, yet misjudge it. An agent that does well here can be trusted to find the paper that meets a need, or to say that none does.

Looking ahead, we envision two directions. One is a harness for literature search, guided by where each agent fails and which papers it finds hard. The other is a world model of a field, distilled from surveys, so that an agent knows how to search a field and check a condition before the search begins. We hope COMPLIT serves as a foundation for future work on scientific literature search.

Limitations

COMPLIT draws its constraints from comparison tables in arXiv surveys from English arXiv papers and relies on the distinctions survey authors chose to compare. Constraints and queries are generated and verified automatically, but we check every constraint and every query by hand to remove remaining errors. Finally, open-web agents search a live web, so their results may shift slightly if the experiments are rerun.

References

- Anirudh Ajith, Mengzhou Xia, Alexis Chevalier, Tanya Goyal, Danqi Chen, and Tianyu Gao. 2024. Lit-search: A retrieval benchmark for scientific literature search. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15068–15083.
- Mohammed Ali, Abdelrahman Abdallah, Amit Agarwal, Hitesh Laxmichand Patel, and Adam Jatowt. 2026. Recor: Reasoning-focused multi-turn conversational retrieval benchmark. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 2688–2723.
- Anthropic. 2025. Claude code: Agentic coding tool. <https://code.claude.com>.
- Akari Asai and Eunsol Choi. 2021. Challenges in information-seeking qa: Unanswerable questions and paragraph retrieval. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1492–1504.
- Jonathan Bragg, Mike D’Arcy, Nishant Balepur, Dan Bareket, Bhavana Dalvi Mishra, Sergey Feldman, Dany Haddad, Jena Hwang, Peter Jansen, Varsha Kishore, et al. 2026. Astabench: Rigorous benchmarking of ai agents with a scientific research suite. In *International conference on learning representations*, volume 2026, pages 110136–110223.
- Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. 2021. A dataset of information-seeking questions and answers anchored in research papers. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4599–4610, Online. Association for Computational Linguistics.
- DeepSeek-AI. 2026. DeepSeek-V4: Towards highly efficient million-token context intelligence. *arXiv preprint arXiv:2606.19348*.
- Yuwen Du, Tian Jin, Jing Kang, Xianghe Pang, Jingyi Chai, Tingjia Miao, Fenyi Liu, WenHao Wang, Yuzhi Zhang, Siheng Chen, et al. 2026. Towards self-evolving agentic literature retrieval. In *ICML 2026 AI for Science Workshop*.
- GLM-5-Team. 2026. *Glm-5: from vibe coding to agentic engineering*. *Preprint*, arXiv:2602.15763.
- Steven A Greenberg. 2009. How citation distortions create unfounded authority: analysis of a citation network. *Bmj*, 339.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.
- Dany Haddad, Dan Bareket, Joseph Chee Chang, Jay DeYoung, Jena D Hwang, Uri Katz, Mark Polak, Sangho Suh, Harshit Surana, Aryeh Tiktinsky, et al. 2026. Understanding usage and engagement in ai-powered scientific research tools: The asta interaction dataset. *arXiv preprint arXiv:2602.23335*.
- Yichen He, Guanhua Huang, Peiyuan Feng, Yuan Lin, Yuchen Zhang, Hang Li, et al. 2025. Pasa: An llm agent for comprehensive academic paper search. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11663–11679.
- Tiansheng Hu, Yilun Zhao, Canyu Zhang, Arman Cohan, and Chen Zhao. 2026. Sage: Benchmarking and improving retrieval for deep research agents. *arXiv preprint arXiv:2602.05975*.
- Pengcheng Jiang, Zhiyi Shi, Kelly Hong, Xueqiang Xu, Jiashuo Sun, Jimeng Sun, Hammad Bashir, and Jiawei Han. 2026. Harness-1: Reinforcement learning for search agents with state-externalizing harnesses. *arXiv preprint arXiv:2606.02373*.
- Kimi Team. 2026. Kimi K3: Open frontier intelligence. *arXiv preprint arXiv:2607.24653*.
- Polina Kirichenko, Mark Ibrahim, Kamalika Chaudhuri, and Samuel J Bell. 2026. Abstentionbench: Reasoning llms fail on unanswerable questions. *Advances in Neural Information Processing Systems*, 38.
- Carol C. Kuhlthau. 1991. Inside the search process: Information seeking from the user’s perspective. *Journal of the American Society for Information Science*, 42(5):361–371.
- Philippe Laban, Hiroaki Hayashi, Yingbo Zhou, and Jennifer Neville. 2026. LLMs get lost in multi-turn conversation. In *International Conference on Learning Representations*, volume 2026, pages 54738–54778.
- Zhuofeng Li, Haoxiang Zhang, Cong Wei, Pan Lu, Ping Nie, Yi Lu, Yuyang Bai, Shangbin Feng, Hangxiao Zhu, Ming Zhong, et al. 2026. Beyond semantic similarity: Rethinking retrieval for agentic search via direct corpus interaction. *arXiv preprint arXiv:2605.05242*.

- Kyle Lo, Lucy Lu Wang, Mark Neumann, Rodney Kinney, and Daniel S Weld. 2020. S2orc: The semantic scholar open research corpus. In *Proceedings of the 58th annual meeting of the association for computational linguistics*, pages 4969–4983.
- Xuan Lu, Sifan Liu, Bochao Yin, Yongqi Li, Xinghao Chen, Hui Su, Yaohui Jin, Wenjun Zeng, and Xiaoyu Shen. 2025. Multiconir: Towards multi-condition information retrieval. In *EMNLP (Findings)*, pages 13471–13494.
- OpenAI. 2025. Deep research system card. <https://openai.com/index/deep-research-system-card/>.
- OpenAI. 2026a. GPT-5.6: Frontier intelligence that scales with your ambition. <https://openai.com/index/gpt-5-6/>.
- OpenAI. 2026b. GPT-5.4 thinking system card. <https://openai.com/index/gpt-5-4-thinking-system-card/>.
- Vedrana Pavlovic, Tracey Weissgerber, Dejana Stanisavljevic, Tatjana Pekmezovic, Ognjen Milicevic, Jelena Milin Lazovic, Andja Cirkovic, Marko Savic, Nina Rajovic, Pavle Piperac, et al. 2021. How accurate are citations of frequently cited papers in biomedical literature? *Clinical Science*, 135(5):671–681.
- Qwen Team. 2026. Qwen3.5: Towards native multi-modal agents.
- Pranav Rajpurkar, Robin Jia, and Percy Liang. 2018. Know what you don’t know: Unanswerable questions for squad. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 784–789.
- Stephen Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and trends® in information retrieval*, 4(1-2):1–174.
- Rulin Shao, Akari Asai, Shannon Zejiang Shen, Hamish Ivison, Varsha Kishore, Jingming Zhuo, Xinran Zhao, Molly Park, Samuel G. Finlayson, David Sontag, Tyler Murray, Sewon Min, Pradeep Dasigi, Luca Soldaini, Faeze Brahman, Wen-tau Yih, Tongshuang Wu, Luke Zettlemoyer, Yoon Kim, and 2 others. 2026. DR Tulu: Reinforcement learning with evolving rubrics for deep research. In *International Conference on Machine Learning (ICML)*.
- Rulin Shao, Rui Qiao, Varsha Kishore, Niklas Muenighoff, Xi Victoria Lin, Daniela Rus, Bryan Kian Hsiang Low, Sewon Min, Wen-tau Yih, Pang Wei Koh, et al. 2025. Reasonir: Training retrievers for reasoning tasks. *arXiv preprint arXiv:2504.20595*.
- Amanpreet Singh, Mike D’Arcy, Arman Cohan, Doug Downey, and Sergey Feldman. 2023. SciRepEval: A multi-format benchmark for scientific document representations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*.
- Zeinab Sadat Taghavi, Ali Modarressi, Yunpu Ma, and Hinrich Schütze. 2025. Impliret: Benchmarking the implicit fact retrieval challenge. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 33156–33178.
- Diane Tchuindjo, Devavrat Shah, and Omar Khattab. 2026. Obliq-bench: Exposing overlooked bottlenecks in modern retrievers with latent and implicit queries. *arXiv preprint arXiv:2605.06235*.
- Pertti Vakkari. 2001. A theory of the task-based information retrieval process: a summary and generalisation of a longitudinal study. *Journal of documentation*, 57(1):44–60.
- Jianyou Andre Wang, Kaicheng Wang, Xiaoyue Wang, Prudhviraj Naidu, Leon Bergen, and Ramamohan Paturi. 2023. Scientific document retrieval using multi-level aspect-based queries. *Advances in Neural Information Processing Systems*, 36:38404–38419.
- Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. 2025. Browsecomp: A simple yet challenging benchmark for browsing agents. *arXiv preprint arXiv:2504.12516*.
- Lei Xiong, Kun Luo, Ziyi Xia, Wenbo Zhang, Jin-Ge Yao, Zheng Liu, Jingying Shao, Jianlyu Chen, Hongjin Qian, Xi Yang, et al. 2026. Autoresearchbench: Benchmarking ai agents on complex scientific literature discovery. *arXiv preprint arXiv:2604.25256*.
- Dongkeun Yoon, Seungone Kim, Sohee Yang, Sunkyoung Kim, Soyeon Kim, Yongil Kim, Eunbi Choi, Yireun Kim, and Minjoon Seo. 2025. Reasoning models better express their confidence. *Advances in Neural Information Processing Systems*, 38:103869–103896.
- Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*.

A Benchmark Construction

A.1 Benchmark Statistics

Overview. COMPLIT draws on 193 survey papers that contain at least one comparison table over the papers they review. Table 4 summarizes each set. The four sets cover 443 distinct target papers. STANDARD and IMPLICIT share no target paper, and CUMULATIVE reuses the 220 queries of IMPLICIT. UNMET shares 142 target papers with STANDARD and 41 with IMPLICIT, but no query.

	STD.	IMP.	CUM.	UNM.
Queries or conversations	350	220	220	300
Target papers	251	179	179	196
Surveys	115	117	117	102
Constraints per query	6.9	5.2	5.2	7.8
Turns per conversation	—	—	3.9	—

Table 4: **Overview of the four sets.** Settings are abbreviated as in Table 2. UNMET counts include the injected false constraint.

Constraint types and evidence levels. Figure 6 shows how constraints divide by type and level. Content constraints are the largest type in every set, and figural constraints make up a quarter of STANDARD and two fifths of UNMET. In STANDARD, about half of the content constraints are explicit, while figural constraints spread across all three levels. With comparative constraints counted as inferred, 28.9% of STANDARD constraints are inferred. IMPLICIT draws only from inferred content constraints, inferred figural claims whose fact the paper’s text does not state, and comparative constraints, so its levels are fixed by construction.

Domains, years, and query length. Figure 7 shows the spread of targets and queries. Every set covers all eight arXiv groups. Physics is the largest group in STANDARD and UNMET, computer science in IMPLICIT, and economics the smallest in all three. Target papers range from 1998 to 2026, with a median year of 2020 in STANDARD and 2019 in IMPLICIT. STANDARD queries carry 4 to 17 constraints, while IMPLICIT queries carry 3 to 10, most often four.

Intents. Intents follow the taxonomy of Haddad et al. (2026). We keep the 11 of its 16 purposes that ask for papers and drop the 5 generative ones, such as drafting and ideation. Figure 8 gives their distribution: five purposes account for over 80% of

queries in every set, and IMPLICIT draws more on causal, gap, and citation purposes than STANDARD.

A.2 Human Verification

The authors reviewed every paper in a web interface that shows the paper’s full text and figures next to each generated item. Review proceeds in stages, and a paper enters the benchmark only after it passes the query stage.

1. **Constraints.** For each content, comparative, and figural constraint, the reviewer checks that the recorded evidence appears in the paper and supports the constraint, and corrects its value or level. Verifier-dropped table cells are shown too, so a wrong drop can be restored.
2. **Intents.** The reviewer checks that each intent fits the paper and claims nothing the paper does not support.
3. **Queries.** The reviewer checks that the query identifies the target and that its constraint set matches the text, and edits either one.

Table 5 counts the changes over the 360 papers that passed the query stage of the main review track. Reviewers removed between a fifth and a quarter of the content, comparative, and figural constraints and edited the values of a quarter of the remaining content constraints. They also restored 120 table cells the verifier had dropped and edited 81 of them. Few intents and queries were removed outright. Most query edits changed the constraint set to match the text. IMPLICIT was reviewed on a separate track with the same three checks, where 26 of 400 queries were removed and 86 had their constraint set changed.

Item	Before	Removed	Edited
Content constraints	1,582	304 (19.2%)	317
Comparative constraints	849	215 (25.3%)	18
Figural claims	4,654	851 (18.3%)	1,378
Intents	894	2 (0.2%)	333
Queries	2,480	1 (0.0%)	137

Table 5: **Changes made in human verification**, over the 360 papers of the main review track. *Edited* counts items whose value or text changed; for queries, it counts changed constraint sets. The figural count is net of 228 claims that reviewers split or rewrote.

A.3 CUMULATIVE and UNMET Construction

Scheduler rules. The CUMULATIVE scheduler turns each IMPLICIT query into a conversation un-

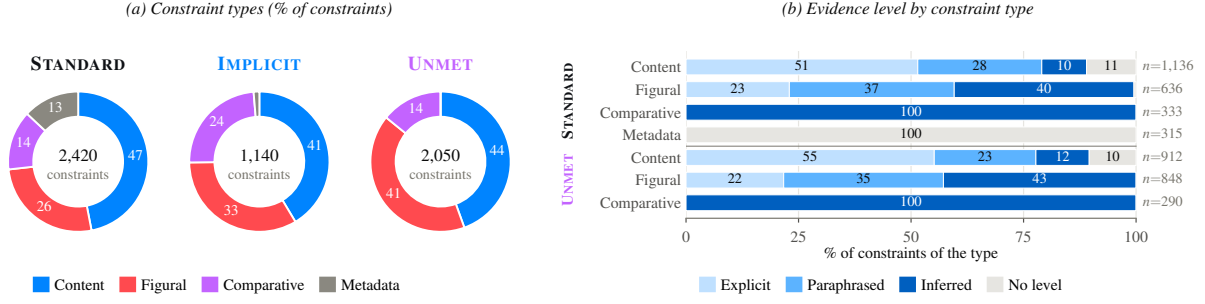


Figure 6: **Constraints in COMPLIT.** (a) Share of each constraint type, over all constraints of a set. CUMULATIVE has the constraints of IMPLICIT, and UNMET excludes the injected constraint. (b) Evidence level of each constraint type in STANDARD and UNMET. Comparative constraints are inferred by definition, and metadata constraints carry no level. *No level* also marks content constraints whose field was renamed during query writing and no longer matches a labeled attribute.

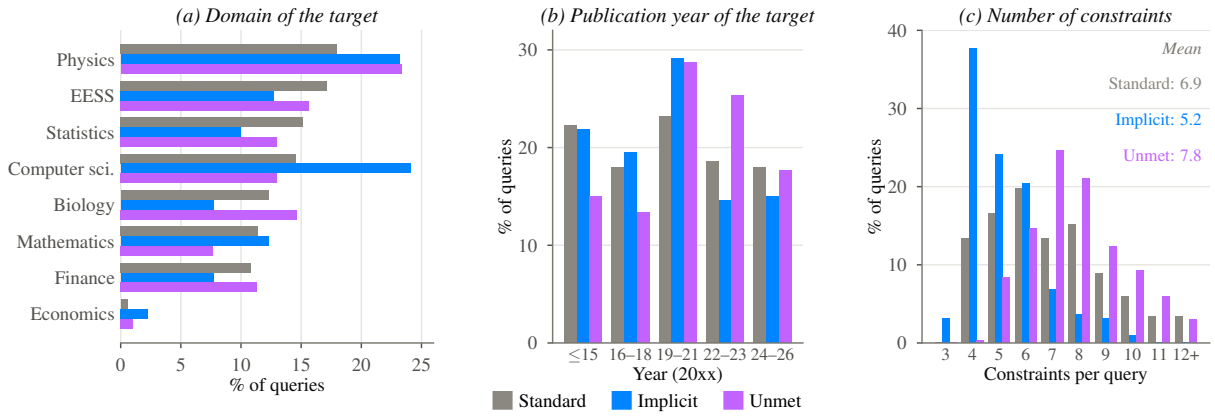


Figure 7: **Targets and queries in COMPLIT.** (a) arXiv group of the target paper’s primary category, where physics pools astro-ph, cond-mat, hep, nucl, quant-ph, and physics. (b) Publication year of the target. (c) Number of constraints per query, where UNMET includes the injected constraint. CUMULATIVE has the distributions of IMPLICIT.

der the following rules, and code enforces every rule except the last.

- The opening turn states the researcher’s broad goal and carries no constraint. Its motivation stays one level more general than every constraint, and the audit caps its score when it shares two or more content words with them.
- Each later turn carries one to three constraints that form one thought, and every constraint appears exactly once.
- A conversation has at least three turns, and the last turn asks the agent to commit to one paper or to say that none qualifies.
- Each turn names a search purpose from the intent catalog, and the purpose may change across turns.
- The scheduler holds back the constraint that most directly identifies the target for a later turn, so

the opening alone does not identify it. This rule is given only as an instruction.

An audit by gpt-5.6-sol rescues each turn’s motivation and asks for a rewrite below 0.8. Figure 9 summarizes the resulting conversations: most have four turns, and most later turns reveal two constraints.

User simulator. At evaluation time, gpt-5.4-mini at temperature 0 delivers the prepared utterance of each turn. When the agent’s last message asks a question or proposes a paper, the simulator adds one sentence that answers from what the user has already said, or that states the project has no requirement on that point. It never adds a constraint, so every agent faces the same constraints at each turn.

UNMET construction. We start from answerable queries that the reference systems answer correctly, so that a failure on UNMET reflects abstention

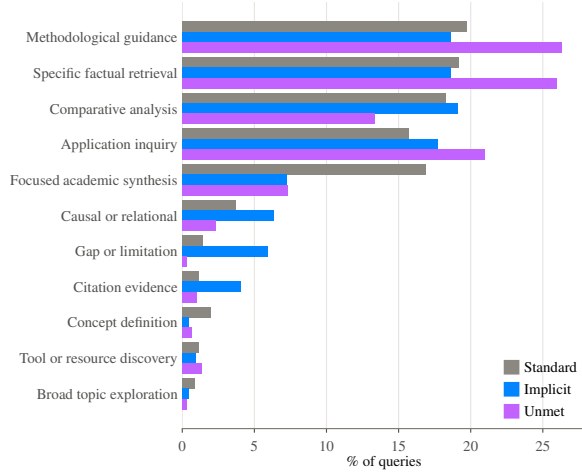


Figure 8: **Intent of each query**, from 11 search purposes of Haddad et al. (2026). CUMULATIVE has the intents of IMPLICIT at the query level.

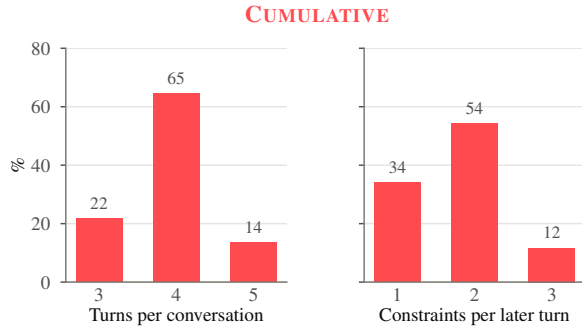


Figure 9: **CUMULATIVE conversations**. Turns per conversation and constraints per turn after the opening, as a share of the 220 conversations and of their 642 later turns.

rather than retrieval. We use only the variant without metadata. For each query, gpt-5.6-sol writes one false clause of one of two kinds.

- **A table value.** The clause asserts a false value on a survey table column that the query does not use. The true value must appear in the paper’s body but not in its title or abstract.
- **A figure fact.** The clause reverses a verified figural claim, such as an ordering or a trend, whose fact the paper’s text also states.

Every original sentence of the query is kept, and the clause adds 7 to 30 words. A judge then reads the target paper and labels the clause as *refuted* when the paper contradicts it, *silent* when the paper does not settle it, or *satisfied*. We drop satisfied clauses.

A.4 Examples

We show one instance per setting, with every constraint, its type, its evidence level, and the evidence that supports it in the target paper (Figures 10–13). Metadata constraints carry no level (§3.2), and a comparative constraint is always *inferred*. The released IMPLICIT records do not store a level for most constraints, so the levels we show for the IMPLICIT example come from our own reading of the paper.

Example of a STANDARD query (stat)

Query. I’m building a drug-target binding affinity predictor and need a clean dual-CNN design that processes compound and protein inputs through separate convolutional streams before merging them. The method I’m implementing requires exactly this two-branch architecture with concatenation prior to the fully connected layers, so I can reproduce the pipeline without wrestling with graph-based or recurrent alternatives. I need the work that uses the same type of ligand input as ‘WideDTA: prediction of drug-target binding affinity’ but a different protein representation from ‘DeepAffinity: Interpretable Deep Learning of Compound-Protein Affinity through Unified Recurrent and Convolutional Neural Networks’. It comes from the late 2010s on arXiv.

Intent. Methodological guidance. **Gold.** *DeepDTA: Deep Drug-Target Binding Affinity Prediction* (arXiv:1801.10193, 2018). **Survey.** *Drug-Target Interaction/Affinity Prediction: Deep Learning Models and Advances Review* (arXiv:2502.15346).

Type	Level	Constraint	Evidence in the target paper
CONTENT	<i>explicit</i>	a CNN encodes the drug	“We proposed a CNN-based prediction model that comprises two separate CNN blocks, each of which aims to learn representations from SMILES strings and protein sequences.” (Proposed Model)
CONTENT	<i>explicit</i>	a CNN encodes the protein	the same sentence
FIGURAL	<i>paraphrased</i>	two streams with separate CNN blocks for compound SMILES and protein sequences	Figure 2, “DeepDTA model with two CNN blocks to learn from compound SMILES and protein sequences.”
FIGURAL	<i>paraphrased</i>	a concatenation merges the two streams before the fully connected layers	Figure 2 draws the concatenation feeding the fully connected layers, and the text states “The final features of the max-pooling layers were concatenated and fed into three FC layers” (Proposed Model).
COMPARATIVE	<i>inferred</i>	the same ligand representation as WideDTA (arXiv:1902.04166)	The survey table records SMILES strings as the ligand input of both papers, with WideDTA also adding substructure words. DeepDTA predates WideDTA and never mentions it.
COMPARATIVE	<i>inferred</i>	a different protein representation than DeepAffinity (arXiv:1806.07537)	The survey table records protein sequences for DeepDTA and structural property strings for DeepAffinity. DeepDTA never mentions DeepAffinity.
METADATA	–	year = 2018; venue = arXiv	arXiv record

Figure 10: **A STANDARD query.** Its eight constraints cover all four types and all three evidence levels. The query names WideDTA and DeepAffinity but not their input representations, so the two comparative constraints require the survey table or the two papers.

Example of an IMPLICIT query (math)

Query. I'm trying to identify parametric spatiotemporal dynamical models where the coefficients exhibit step changes, but I'm struggling with the sharp transitions in the loss landscape when tuning regularization. I need a procedure that handles these coefficient variations without requiring Bayesian uncertainty quantification, and I want to avoid methods that get bogged down in handling missing data. I also need an approach that is illustrated with less real data than 'DeepMoD: Deep learning for Model Discovery in noisy data', so I can focus on the core identification pipeline without that overhead.

Intent. Methodological guidance. **Gold.** *Data-driven identification of parametric partial differential equations* (arXiv:1806.00732, 2018). **Survey.** *A Review of Data-Driven Discovery for Dynamic Systems* (arXiv:2210.10663).

Type	Level	Constraint (survey cell)	Evidence in the target paper
CONTENT	<i>inferred</i>	no Bayesian uncertainty quantification (UQ = no)	The paper never mentions Bayes, posteriors, or uncertainty. It selects one model per method with a point criterion: "we evaluate the models using the AIC-inspired loss function" (§2.3).
CONTENT	<i>inferred</i>	does not handle missing data (missing data = no)	The paper never mentions missing data. Every example is simulated, and "For the noise-free dataset we use the discrete Fourier transform for computing derivatives" (§3.1).
FIGURAL	<i>inferred</i>	the loss stays low over the regularization sweep, then jumps	Figure 2 plots the loss for 50 thresholds and 50 penalties, and in both panels it stays near its minimum and then jumps. The text says only "For sufficiently high values, the model is too sparse and is no longer predictive" (§2.3).
FIGURAL	<i>paraphrased</i>	the coefficients show a step change	The Figure 5 caption states it: "coefficients for Navier-Stokes equations exhibiting jump in Reynolds number from 100 to 75 at $t = 10$."
COMPARATIVE	<i>inferred</i>	less real data than arXiv:1904.09406	The gold is "demonstrated on four canonical models", all simulated, while the other paper is applied to "noisy experimental time-series data from a gel electrophoresis experiment". The gold predates it and never mentions it.

Figure 11: **An IMPLICIT query.** Four of the five constraints must be inferred: the paper never states that it is non-Bayesian or ignores missing data, the jump in the loss is visible only in Figure 2, and the comparative constraint names a property without its value. The step change is stated in the text and in the Figure 5 caption. The loss-curve constraint also separates the gold from Online WSINDy (arXiv:2203.03979), a later method that satisfies the other four constraints but never plots loss against its threshold. The agents' answers to this query appear in Table 16.

Example of a CUMULATIVE conversation (stat)

Gold. *Model-based Differentially Private Data Synthesis and Statistical Inference in Multiply Synthetic Differentially Private Data* (arXiv:1606.08052, 2016). **Survey.** *Comparative Study of Differentially Private Data Synthesis Methods* (arXiv:1602.01063, revised through 2019).

Turn	Revealed	User utterance
1	none (opening)	I'm developing a critique of approaches that protect sensitive information while still allowing useful data release, and I want a concrete limitation to anchor the argument. Can you help me find one paper that provides that case?
2	CONTENT $\times 2$: privacy cost only on sanitizing the posterior; releasing records is free post-processing	The first thing I need is a clean accounting boundary, since I don't want to blame any step that consumes no additional privacy budget. The privacy cost should fall only on sanitizing the posterior, with generation of the released records treated as free post-processing.
3	CONTENT $\times 2$: one route needs an intractable normalizing constant; the other an analytic sensitivity bound	With that boundary in place, the bottleneck has to persist across both natural routes for carrying out the protected inference, so it cannot be dismissed as an implementation quirk. One route must face an intractable normalizing constant, while the other must require an analytic sensitivity bound. Please commit to one paper now, or say plainly if nothing meets everything I've asked for.

Figure 12: **A CUMULATIVE conversation.** The opening states the researcher's goal and field but no constraint. The next two turns reveal two constraints each, and the agent is scored on its answer after the last turn. The gold states that drawing the released records "does not incur any additional privacy cost as it can be regarded as post-processing", notes that the posterior's "normalizing constant ... might not have a close-form expression", and sanitizes sufficient statistics with the Laplace mechanism, which requires their global sensitivity.

Example of an UNMET query (stat)

Query. I need a Python tool that implements a structured four-step causal inference workflow, taking domain knowledge and data to construct causal graphs and estimate effects. **The workflow should identify the target estimand before modeling the causal mechanisms.** It should support standard identification strategies like backdoor, frontdoor, and instrumental variable adjustments, with built-in mechanisms to refute estimates and validate results before deployment.

Gold. $\emptyset$ (abstain). **Near-miss.** *DoWhy: An End-to-End Library for Causal Inference* (arXiv:2011.04216, 2020). **Intent.** Tool and resource discovery. **Perturbation.** A figure fact, reordered.

Type	Level	Constraint	Status in the near-miss
CONTENT	<i>explicit</i>	implemented in Python	satisfied
CONTENT	<i>paraphrased</i>	supports backdoor, frontdoor, and IV identification	satisfied
FIGURAL	<i>paraphrased</i>	a four-step analysis pipeline	satisfied (Figure 1)
FIGURAL	<i>explicit</i>	the four steps are Model causal mechanisms, Identify the target estimand, Estimate causal effect, and Refute estimate	satisfied (Figure 1); not stated in the query, and the injected clause reverses its first two steps
FIGURAL	—	identify the estimand before modeling causal mechanisms (injected)	violated: “DoWhy’s API is organized around the four key steps that are required for any causal analysis: Model, Identify, Estimate, and Refute.”

Figure 13: **An UNMET query.** The injected constraint (bold) reverses the step order that DoWhy’s Figure 1 depicts. The original target thus becomes a near-miss that satisfies every other constraint, and the correct answer is to abstain.

B Lexical Retrieval Across Benchmarks

We compare how hard COMPLIT questions are to match lexically against existing literature-search benchmarks. We run one BM25 retriever over one corpus for six benchmarks and for STANDARD and IMPLICIT.

Setup. The corpus holds the titles and abstracts of all 3,182,867 arXiv papers in the arXiv metadata snapshot of September 28, 2026. Unlike the BM25 baseline of Appendix D.1, it indexes no full bodies, so that the gold papers of every benchmark are in the corpus. BM25 uses tantivy ($k_1=1.2$, $b=0.75$) and takes each question verbatim. Several benchmarks list many gold papers per question, which caps Recall@1 below one, so we report $H@k$, the share of questions with at least one gold paper in the top k .

Gold papers. We map every gold paper to an arXiv id and skip questions left with none. PaSa (He et al., 2025) lists arXiv ids for RealScholarQuery and the AutoScholarQuery test split. For LitSearch (Ajith et al., 2024), 520 of 639 gold papers have an arXiv id in Semantic Scholar, which leaves 498 of 597 questions. For PaperFindingBench (Bragg et al., 2026), we use the May 2025 test split; 392 of 803 listed papers have an arXiv id, and most semantic questions list no gold paper because they are scored by a judge, which leaves 101 of 267 questions. For AutoResearchBench (Xiong et al., 2026), we use the Wide subset. For DORIS-MAE (Wang et al., 2023), we use the BIRCO release, match 5,505 of its 5,544 candidate abstracts to arXiv by text, and count a paper as gold if its relevance score is at least 1.0. DORIS-MAE is thus scored against all of arXiv rather than its 100-paper candidate pools. We exclude LitQA2, since only one of its 199 questions cites an arXiv paper.

Results. Among benchmarks with one or two gold papers per question, IMPLICIT is the hardest to match at the top ranks, while STANDARD is the easiest (Table 6). BM25 ranks a gold paper first for 1.8% of IMPLICIT questions, against 7.1–18.5% on PaSa, PaperFindingBench, and LitSearch. By the top 100 the gap closes, so the gold paper is often retrieved but rarely ranked first: the constraints it satisfies are inferred rather than stated in the question. STANDARD reaches 25.7% at $H@1$, the highest of all benchmarks, consistent with its constraints being stated in the question. PaperFindingBench includes 27 metadata questions on au-

Benchmark	H@1	H@5	H@20	H@100
PaSa (RealScholarQuery)	12.0	34.0	50.0	74.0
PaSa (AutoScholarQuery)	7.1	15.1	25.8	41.2
PaperFindingBench	14.9	26.7	36.6	44.6
LitSearch	18.5	30.9	42.0	56.2
AutoResearchBench (Wide)	14.2	36.0	61.5	84.5
DORIS-MAE	0.0	25.0	62.5	83.9
CompLit STANDARD	25.7	45.4	58.0	74.6
CompLit IMPLICIT	1.8	13.2	27.3	48.2

Table 6: **Lexical retrieval across literature-search benchmarks.** BM25 over title and abstract of a shared corpus of 3.18M arXiv papers. $H@k$ is the share of questions with at least one gold paper in the top k .

thors and years, which a title-and-abstract index cannot answer.

C Does the Agent Harness Matter?

Claude Code and Codex CLI run in their own harnesses, while Kimi and DeepSeek run in inspect_evals. We test whether the harness, rather than the model, explains part of the gap, by running DeepSeek on IMPLICIT inside the Claude Code and Codex CLI harnesses.

Setup. The model, tools, and prompt are those of the inspect_evals run (Appendix D.1): DeepSeek-V4-Pro, the same arxiv_search, text-only read_paper, and Serper web_search exposed as MCP tools, and the same single user prompt. The built-in web tools of both harnesses are disabled. Each harness keeps its own system prompt, agent loop, and context management. Without a limit, DeepSeek keeps searching and rarely answers, so each harness allows 30 tool calls per query, after which further calls are blocked with an instruction to answer. In inspect_evals, a query ends after 30 messages (median 19 tool calls), and runs that have not committed receive a forced final answer.

Results. The two harnesses score the same, and both match inspect_evals once an answer is forced (Table 7). DeepSeek reaches 41.4% in Claude Code and 40.9% in Codex CLI, against 37.3% in inspect_evals with the forced final answer. Without it, inspect_evals reaches 12.7%, so the gap comes from forcing a commitment, not from the harness. The remaining 4pp coincides with the larger tool budget of the harnesses (30 against a median of 19 calls). In the same harnesses, DeepSeek stays far below Claude Code and Codex CLI with their own models (65.9% and

Harness	Forced answer	Tool calls	Acc.
inspect_evals	–	19	12.7
inspect_evals	✓	19	37.3
Claude Code	✓	31	41.4
Codex CLI	✓	30	40.9

Table 7: **DeepSeek-V4-Pro on IMPLICIT in three harnesses.** Model, tools, and prompt are fixed. Tool calls is the median per query. In inspect_evals, the answer is forced after the 30-message limit; in Claude Code and Codex CLI, after 30 tool calls.

67.7%, Table 2), so the harness does not explain that gap.

C.1 Prompt Templates

We list the core prompts for constraint derivation and query writing, verbatim, except that we cut passages not needed to follow them and mark each cut as [... omitted ...]. Placeholders in braces are filled per paper. We omit the verifier and retry prompts that accompany each step: every verifier scores the generator’s output against the same inputs, and a retry appends the verifier’s issues to the generator prompt. Each step runs Qwen3.5-122B-A10B at temperature 0.3 with the system message “*You are a helpful research assistant. Respond only with valid JSON.*”

Content constraints: the grounder. Given the survey’s table cells for the target paper and up to 60,000 characters of the paper’s body, the grounder gives each cell a readable attribute name, keeps its value, quotes the supporting passage, labels its evidence level, and keeps or drops it. The labels *literal*, *semantic*, and *inferential* correspond to explicit, paraphrased, and inferred.

Prompt for the Grounder (content constraints and their evidence levels)

```
### Task
You are validating the row of a survey-paper comparison table that describes ONE specific paper. You have:
- the row's `(attribute_key, attribute_value)` pairs (extracted by an earlier step from the survey table cell text), and
- the paper's own title, abstract, and section bodies.

For each `(attribute_key, attribute_value)`, return these fields:

1. `refined_key` -- a more specific, snake_case key that names the actual property this cell encodes for THIS paper, based on the paper's content. Examples:
- "Scale" with value "50 clients" -> refined_key: "number_of_clients"
[... omitted ...]

2. `refined_value` -- the ORIGINAL cell value, preserved FAITHFULLY. Apply ONLY trivial surface normalization: strip surrounding reference numbers like "[32]", trim whitespace, and repair obvious formatting artifacts (e.g. a subscript split by a space, "LiNbO 3" -> "LiNbO3").

You MUST NOT change what the value means. [... omitted ...]
refined_value must mean exactly what the survey-table cell meant.

3. `verdict` -- exactly one of:
- "keep" -- the paper supports the value (literally, semantically, or inferentially), OR the key is a metadata field. Classify the support type in `match_type` below.
- "drop_proper_noun" -- value is just a proper noun (model/dataset/ benchmark name) with no descriptive content.
- "drop_unverified" -- the cell text was generic ("-", "N/A", "yes", "no", "✓", a bare number that is clearly a citation reference, or otherwise empty), OR the paper neither states nor lets you derive the value via reasoning of any depth.
[... omitted ...]

4. `match_type` -- REQUIRED when verdict = "keep" (omit / null otherwise).
Exactly one of:
- "metadata" -- `refined_key` is one of: year, venue, publication date, authors, affiliations, citation count, paper id, reference number. [... omitted ...]
- "literal" -- paper states the SAME value with at most trivial surface differences: identical wording, identical numbers, casing/whitespace, OR an abbreviation vs its expanded form (e.g. "OOK" <-> "on-off keying"). [... omitted ...]
- "semantic" -- paper states the same concept with DIFFERENT wording that requires a one-step lexical or conceptual substitution: paraphrase, true synonym, hypernym/hyponym, stylistic rewording.
- "inferential" -- value is not stated, but can be DERIVED from the paper's method, architecture, setup, results, or from the paper's stated limitations/constraints via reasoning. [... omitted ...]

Worked example A -- cell `refined_value = "O(N) complexity"`:
- paper says "the algorithm runs in O(N) time" -> literal
- paper says "the algorithm has linear complexity" -> semantic (one-step substitution: linear <-> O(N))
- paper describes a single pass over the input with constant work per token, no big-O statement -> inferential (derive from method description)
- paper says nothing about runtime/scaling -> drop_unverified
[... omitted ...]

5. `evidence` -- for "keep", quote a snippet (<= 200 chars) from the paper text above that supports the value. The quoted portion MUST be a VERBATIM substring of the paper content. [... omitted ...] For "drop", briefly explain why instead.

### Paper content
{paper_text}

### Comparison table row
- Table title: {table_title}
- Mention: {mention}
- Paper title: {paper_title}

### Surrounding context from the SURVEY (around this table)
[... omitted ...]
{surrounding_context}

### Original attribute_values
```

```
{attribute_values_text}

[... omitted ...]

### Output (JSON only, no prose):
{
  "attributes": [
    {
      "original_key": "Scale",
      "original_value": "50 clients",
      "refined_key": "number_of_clients",
      "refined_value": "50 clients",
      "verdict": "keep",
      "match_type": "literal",
      "evidence": "Section 4.2: 'we evaluate with 50 participating clients'"
    },
    [... omitted ...]
  ]
}
```

Comparative constraints. Given the whole comparison table, the LLM writes two to five facts that each distinguish the target from one named sibling paper on one column.

Prompt for Comparative Constraints

```
### Task:
Write 2-5 pairwise comparison facts about the row marked `<- TARGET`. Each fact compares the TARGET with exactly one sibling
paper from the table along a single dimension, naming the sibling by its exact title in quotes.

### Table:
{comparison_block}

### Shape:
"<TARGET verb-phrase> <directional word> '<Sibling Paper Title>'"

Directional words pair with `as` or `than` so the sibling is the reference point for the comparison: **same ... as**, **larger /
higher / more / newer / longer ... than**, **smaller / lower / fewer / older / shorter ... than**, **different ... from**.

### Examples (good):
- "uses the same batch size as 'Accurate, Large Minibatch SGD: Training ImageNet in 1 Hour'"
[... omitted ...]
### Rules:
1. Exactly one sibling per fact -- split multi-sibling comparisons into separate facts.
2. The sibling title must be **load-bearing** -- removing it from the `phrasing` should break the clause. "Shares X with
'Sibling'" and "differs from 'Sibling' which uses Y" are decorative (they still make sense without the sibling); skip them.
3. Do not leak the TARGET's concrete value into `phrasing`; the searcher recovers it by looking up the sibling.
4. Skip dimensions where the TARGET's value is missing / blank / "-" / "N/A", or whose values are bare reference numbers (e.g.
"51", "70").
5. Pairwise comparisons only -- no survey-wide superlatives or ranks ("the largest", "ranks 3rd", "the earliest listed", etc.).

[... omitted ...]
### Output Format (JSON only):
{
  "multi_doc_constraints": [
    {
      "dimension": "<column name from the table>",
      "target_value": "<TARGET's concrete value in that column>",
      "referenced_title": "<exact sibling title as shown above>",
      "referenced_value": "<that sibling's value in the same dimension>",
      "relation": "same | larger | smaller | older | newer | more | fewer | shorter | longer | higher | lower",
      "phrasing": "<short English clause naming the sibling by title>",
      "confidence": 0.0
    }
  ]
}
[... omitted ...]
If no cross-paper comparison is defensible, return `{ "multi_doc_constraints": [] }`.
```

Figural constraints. The VLM receives one figure with its caption and writes at most three claims about what the figure shows. A few papers were processed with other Qwen3.5 sizes.

Prompt for Figural Claims (VLM)

```
### Task
Analyze this figure from the academic paper "{paper_title}".

### Figure caption
{caption}
```

```

### Instructions
1. First, classify this figure into one of these categories:
  - "results_plot": Charts or graphs showing experimental results, metrics, or performance comparisons
  - "architecture": Model architecture diagrams, system overviews, pipeline diagrams
[... omitted ...]
2. If the category is NOT "uninformative", extract information that a researcher would find useful when searching for this paper:

  **Quantitative** (from results plots/tables -- focus on RANKINGS, not exact numbers):
  - Which method ranks best/worst on which dataset or metric
  - Relative comparisons ("outperforms all baselines on X")
[... omitted ...]
  **Qualitative** (from architecture/method figures):
  - Architecture type (e.g., "encoder-decoder", "two-stream network", "U-Net variant")
  - Pipeline stages and their purpose
[... omitted ...]
Only extract claims clearly supported by the figure and caption.
Skip trivial or obvious observations.
Extract AT MOST 3 claims in total (quantitative + qualitative combined); keep only the most search-relevant, distinctive ones.

### Output (JSON only)
{
  "category": "one of: results_plot, architecture, method_illustration, comparison, uninformative",
  "quantitative": [
    {"claim": "description of ranking or comparison", "metric": "metric name", "rank": "1st/2nd/best/worst", "context": "dataset or setting"}
  ],
  "qualitative": [
    {"claim": "description of method characteristic", "aspect": "architecture/pipeline/component/modality"}
  ]
}

If category is "uninformative", return empty lists for both quantitative and qualitative.

```

Query writing. Given an intent and the paper’s verified constraints, the LLM selects the constraints that serve the intent, writes the request, and records the constraints it used.

Prompt for Query Writing

```

### Task:
Generate a paper-search question whose ANSWER is the TARGET paper below.
The question must read like a real researcher's query -- NOT a filter list -- and must NOT reuse the paper's DISTINCTIVE title phrasing [... omitted ...]

### Target paper:
- Title [... omitted ...]: "{title}"
- Year: {year}
- Venue: {venue}
{enriched_info}
{intention_section}

### Comparison table context:
{table_context}
{vision_info}
{multi_doc_info}

### HARD REQUIREMENTS -- every one MUST hold or the question is rejected:
1. NO distinctive title phrasing (specific multi-word names, coined terms, model names) appears in the question; [... omitted ...]
2. At least 4 constraints total, with at least 2 of type `content`.
3. The constraints TOGETHER uniquely identify the target among the OTHER papers in the table (not just the target alone).
4. Self-contained: no bare acronyms, no references to table columns/dimensions, no outside context -- understandable from the question text alone.
[... omitted ...]

### Build the question from MOTIVATION (M) + CONSTRAINTS (C):
M = the researcher's reason for searching (the intention above). C = the distinguishing facts. Write the most NATURAL-READING question that a person with THIS motivation would actually type. [... omitted ...]

### Constraint sources: each `type` draws ONLY from its own block:
1. **content**: values from the TARGET row of the comparison table only. [... omitted ...]
2. **vision**: claims from the figures/diagrams block only (architecture characteristics, relative rankings from result plots, pipeline structure).
3. **multi_doc**: phrasings from the pre-extracted cross-paper block only. [... omitted ...] do NOT restate the target's own value for that axis ANYWHERE in the question [... omitted ...]
4. **metadata**: year / venue / author affiliation from the target header and enriched info only. [... omitted ...]

### Output Format (JSON only):
{"question": "question.", "constraints": [{"type": "content", "field": "batch_size", "value": "32k"}, [... omitted ...]], [... omitted ...]}

```

D Experimental Details

D.1 Baseline Configurations

We describe the configuration of every baseline. All runs took place between August 12 and August 24, 2026. Web agents searched the live web in that window.

Corpus retrieval. Every retriever returns a top-100 ranking over the 2.95M-paper corpus, and we score its first paper (Recall@1). BM25 indexes titles and full bodies with tantivy ($k_1=1.2$, $b=0.75$). SPECTER2 embeds the title and the first 2,000 characters of the body. ReasonIR and Qwen3-Embed-8B embed up to 32,512 tokens per paper, and Qwen3-Embed-8B uses the instruction “*Given a web search query, retrieve relevant passages that answer the query*”. Qwen3-Rerank-8B reranks the top 100 of Qwen3-Embed-8B using each paper’s title and opening, up to 4,096 tokens.

Corpus search agents. DCI-Agent-Lite runs gpt-5.4-nano with high thinking effort and read and bash tools over per-month files of the arXiv metadata snapshot, which hold titles, authors, abstracts, and categories, for at most 100 turns and one hour per query. Harness-1 is served with vLLM and searches the corpus with a hybrid of BM25 and Qwen3-Embed-8B, with grep, document reading, and pruning tools. It runs a single rollout with at most 50 steps and a 32,768-token context, at temperature 1.0.

Open-web agents. DR Tulu and PaSa run their released pipelines. DR Tulu-8B uses Google search through Serper, Semantic Scholar snippet search, and a page browser summarized by Qwen3-8B, with at most 10 tool calls. PaSa’s crawler generates five search queries and then expands two citation layers, with at most 20 papers per layer. Kimi and DeepSeek run in inspect_ai as a ReAct agent with no system message, provider-default decoding, and three tools:

- `arxiv_search` queries the arXiv API and returns titles, ids, years, authors, and truncated abstracts;
- `web_search` queries Google through Serper and returns titles, URLs, and snippets;
- `read_paper` returns a paper’s full text and attaches up to four figures as images. DeepSeek reads no images.

Claude Code and Codex CLI use their built-in web search and receive `arxiv_search` and

`read_paper` as MCP tools. Claude may also call WebFetch, and Codex a shell restricted to reading files. They have no turn limit and a wall-clock limit of 900 seconds per query.

D.2 Corpus and Survey Access

Each query is built from a survey’s comparison table, so its source survey lists the target with its attributes. We remove these surveys from the corpus for index-based search (§3.4), but open-web systems can still reach them. We therefore count how often the four frontier agents open the source survey of a query, either with `read_paper` or by fetching its page.

Across the four settings, the frontier agents open the source survey in 3.8% of runs (164 of 4,360): 1.8% for Codex, 4.3% for Claude, 4.9% for Kimi, and 4.0% for DeepSeek. Opening it does not help. Accuracy on queries where the agent opens the source survey is lower than on the rest in 10 of the 16 agent and setting pairs, and each of the other six rests on 12 or fewer such queries. This suggests that agents turn to the survey when a query is already hard, and that it does not lead them to the target.

E Additional Results

This section extends the analysis of §4.3 and §4.4 to every baseline with per-query results. These are the four frontier agents, the two corpus search agents, DR Tulu, and PaSa. The corpus retrievers are omitted because only their aggregate scores were kept.

E.1 Accuracy by Domain and Year

Table 8 extends Table 3 to eight baselines and to IMPLICIT. On STANDARD, the pattern of the four frontier agents extends to the other baselines. Every baseline except DeepSeek is least accurate on papers from 2015 or earlier, and physics is the hardest domain for six of the eight. On IMPLICIT, the domain ranking changes: mathematics becomes the hardest domain for Claude and Codex, both at 48.1%, while statistics becomes one of their easiest.

E.2 Accuracy by Constraint Type and Evidence Level

Table 9 splits STANDARD accuracy by the constraints a query contains. For seven of the eight baselines, queries with a comparative constraint are the hardest, and for seven of the eight, queries with a figural constraint are the easiest. Whether a query contains any inferred constraint matters little, but

	STANDARD									IMPLICIT								
	<i>n</i>	DCI	H-1	Tulu	PaSa	DS	Kimi	Claude	Codex	<i>n</i>	DCI	H-1	Tulu	PaSa	DS	Kimi	Claude	Codex
<i>Domain</i>																		
Physics	63	46.0	57.1	15.9	22.2	60.3	68.3	79.4	87.3	51	15.7	33.3	3.9	7.8	29.4	49.0	72.5	66.7
EESS	60	53.3	75.0	26.7	41.7	76.7	80.0	83.3	88.3	28	32.1	39.3	7.1	10.7	28.6	39.3	50.0	64.3
Statistics	53	58.5	75.5	18.9	43.4	66.0	67.9	90.6	84.9	22	27.3	36.4	13.6	22.7	68.2	59.1	90.9	81.8
Computer science	51	54.9	64.7	37.3	41.2	70.6	84.3	84.3	80.4	53	22.6	39.6	3.8	22.6	35.8	47.2	62.3	64.2
Biology	43	58.1	76.7	23.3	58.1	83.7	81.4	90.7	88.4	17	23.5	64.7	5.9	11.8	47.1	70.6	64.7	82.4
Mathematics	40	55.0	62.5	17.5	42.5	80.0	77.5	82.5	87.5	27	29.6	59.3	11.1	14.8	25.9	48.1	48.1	48.1
Finance & economics	40	57.5	80.0	27.5	60.0	92.5	92.5	95.0	95.0	22	27.3	40.9	13.6	22.7	45.5	68.2	77.3	81.8
<i>Publication year</i>																		
≤2015	78	44.9	57.7	11.5	28.2	71.8	65.4	78.2	79.5	48	22.9	39.6	10.4	14.6	31.2	47.9	79.2	72.9
2016–18	63	52.4	77.8	17.5	34.9	63.5	73.0	84.1	87.3	43	23.3	48.8	7.0	18.6	41.9	60.5	62.8	74.4
2019–21	81	65.4	79.0	27.2	54.3	81.5	87.7	96.3	92.6	64	25.0	46.9	6.2	10.9	37.5	51.6	56.2	67.2
2022–23	65	50.8	64.6	30.8	50.8	80.0	81.5	89.2	93.8	32	31.2	40.6	6.2	18.8	43.8	50.0	84.4	75.0
2024–26	63	57.1	69.8	33.3	44.4	73.0	82.5	81.0	82.5	33	18.2	30.3	6.1	21.2	33.3	48.5	51.5	45.5

Table 8: **Accuracy (%) by domain and publication year of the target, for every baseline with per-query results.** H-1 is Harness-1, Tulu is DR Tulu, and DS is DeepSeek. Darker cells mark higher accuracy, with the bins of Table 3.

the share of inferred constraints does. When at least half of a query’s labeled constraints are inferred, the frontier agents lose 5.7 to 10.1 points against queries with no inferred constraint, and PaSa loses 13.0. This is the effect IMPLICIT isolates at full strength, where every constraint is inferred.

	<i>n</i>	DCI	H-1	Tulu	PaSa	DS	Kimi	Cl.	Cx.
<i>Contains a constraint of type</i>									
Content	349	54.4	69.6	23.8	42.7	74.5	77.9	86.0	87.1
Figural	246	59.3	78.0	25.2	50.0	79.3	85.4	91.9	91.9
Metadata	178	56.2	71.9	30.3	44.4	75.8	80.3	87.6	86.0
Comparative	168	48.8	66.1	22.0	42.9	69.0	71.4	83.9	83.3
<i>Share of inferred constraints</i>									
0	136	51.5	68.4	17.6	44.1	72.8	75.0	86.0	90.4
(0, 0.5)	153	58.2	73.2	29.4	45.8	79.1	84.3	88.2	86.9
[0.5, 1)	61	50.8	63.9	23.0	31.1	65.6	68.9	80.3	80.3
<i>Number of constraints</i>									
4	47	53.2	57.4	14.9	27.7	70.2	68.1	70.2	80.9
5–6	127	56.7	65.4	18.9	40.2	74.0	78.0	86.6	88.2
7–8	100	48.0	77.0	29.0	47.0	73.0	78.0	87.0	90.0
9+	76	59.2	75.0	30.3	50.0	78.9	84.2	93.4	85.5

Table 9: **STANDARD accuracy (%) by the constraints a query contains.** The share of inferred constraints is over content and figural constraints with a level label. Cl. is Claude and Cx. is Codex.

Queries with nine or more constraints are easier than queries with four for every baseline in the table, although accuracy does not rise monotonically in between. Claude rises from 70.2% on queries with four constraints to 93.4% on queries with nine or more.

E.3 CUMULATIVE by Turn

Table 10 gives the numbers behind Figure 4. Codex commits late: at the third turn it answers the gold in only 17.3% of conversations, against 37.3% for Claude, but it ends 11 points higher. Every agent is more accurate on longer conversations.

Agent	On gold after turn					Ever another	Recovered
	1	2	3	4	5		
Codex	0.0	8.6	17.3	53.6	61.8	41.4	3.6
Claude	0.0	20.0	37.3	49.1	50.5	65.9	25.5
Kimi	0.5	22.3	32.3	35.9	37.3	74.1	12.3
DeepSeek	0.0	18.6	32.3	36.8	37.7	73.6	11.8
Final, by turns							
	3	4	5				
Codex	56.2	62.0	70.0				
Claude	37.5	52.8	60.0				
Kimi	31.2	38.0	43.3				
DeepSeek	29.2	34.5	50.0				

Table 10: **CUMULATIVE by turn (%).** *On gold after turn* carries a finished conversation’s last answer forward. *Ever another* is the share that chose a non-gold paper at some turn, and *Recovered* the share that did so but ended on the gold. *Recovered* differs from the escaped band of Figure 4, which also counts conversations that leave a wrong paper without ending on the gold. The lower block gives final accuracy for conversations of 3, 4, and 5 turns ($n=48, 142, 30$).

E.4 What Agents Answer on UNMET

Table 11 splits each baseline’s UNMET answers into abstentions, the near-miss, and another paper. When they do not abstain, all baselines except DR Tulu name the near-miss more often than all other papers combined. Codex abstains more often when the paper contradicts the injected clause than when it leaves the clause unsettled, 55.2% against 40.4%. For the other three agents the two rates differ by less than 8 points.

E.5 Cost

Table 12 reports the mean cost per query of the frontier agents. IMPLICIT and CUMULATIVE take more tool calls and time than STANDARD for every agent. Codex is the fastest and makes the fewest

Baseline	Answer			Abstained, by clause	
	Abst.	Near	Other	Ref.	Sil.
Harness-1	9.0	67.7	23.3	–	–
DR Tulu	14.3	29.7	56.0	–	–
PaSa	8.7	54.3	37.0	–	–
DeepSeek	17.3	75.3	6.7	17.7	15.4
Kimi	18.7	76.0	5.3	17.3	25.0
Claude	26.3	62.0	11.7	26.6	25.0
Codex	52.7	44.7	2.7	55.2	40.4

Table 11: **UNMET answers (%)**. *Abst.* is accuracy, *Near* names the near-miss, and *Other* names another paper. On the right, abstention for clauses the paper refutes or leaves silent. DeepSeek’s remaining 0.7% gave no answer. For Harness-1, which returns up to five papers, *Near* and *Other* refer to its first paper.

tool calls, while Kimi spends the most time in every setting.

Agent	STD.	IMP.	CUM.	UNM.
Codex	6.9 / 44	9.5 / 72	11.4 / 125	5.9 / 37
Claude	8.6 / 135	15.5 / 305	13.0 / 446	9.4 / 152
Kimi	10.0 / 237	16.4 / 371	18.1 / 643	12.1 / 228
DeepSeek	13.6 / 143	17.7 / 192	30.0 / 395	16.7 / 199

Table 12: **Mean cost per query**, as tool calls / seconds. Mean dollar cost is \$0.39, \$0.61, \$0.73, and \$0.32 per setting for Codex, computed from its tokens at list price, and \$0.64, \$1.10, \$1.81, and \$0.59 for Claude, as reported by Claude Code.

E.6 Qualitative Comparison of Agents

Tables 13–20 put the four frontier agents side by side on eight queries from six domains, two for each setting. *Calls* counts tool calls, and *Opened* records whether the agent read the gold paper, or the near-miss on UNMET. Excerpts are verbatim from the agents’ answers, and we checked every wrong answer against the text of the paper it names. We chose these cases to illustrate how agents differ, among queries where the agents disagree and every wrong answer demonstrably violates a constraint; they are not a random sample, and aggregate accuracy is given in Table 2.

STANDARD: checking the one distinctive requirement. In Table 13 (physics), the distinctive requirement is a dipolar propagator from opposite dislocations. Codex and DeepSeek open the gold and confirm it. Claude searches for 48 calls, passes over the gold in one search result, settles on a recent model of deformed crystals, and concedes that the paper does not use that phrase. Kimi finds no arXiv match and names a 1994 activated model, crediting it with the gold’s features. In Table 14

(stat), Codex and Claude open the gold and check the sparsity condition, while Kimi and DeepSeek note the gold in their search results, then follow a related method and never open it.

IMPLICIT: checking what must be inferred. In Table 15 (q-bio), whether a candidate’s model has four parameters, is deterministic, and adds a linear final trend must be checked against its model equations. DeepSeek opens the gold and quotes the linear-trend equation. Codex returns a paper whose Richards curve has five parameters and calls its forecasts deterministic, although that paper models the counts as a stochastic process, and Kimi names a paper in which it never found the linear term. Claude runs out of time without opening the gold, which its searches return ten times. In Table 16 (math), the gold never says that it is non-Bayesian, ignores missing data, or uses only simulated data, and the sharp jump in its loss appears only in a figure. Codex and Claude open the gold and check each point, and Claude cites the figure. DeepSeek returns a paper that meets the other constraints but never tunes a regularization parameter and plots no loss curve, and Kimi reaches the message limit without an answer.

CUMULATIVE: revising an earlier answer, or keeping it. In Table 17 (eess), only the last turn reveals the constraints that single out the gold. Claude commits to RadioDUN at the third turn, then opens the gold at the fourth and explicitly drops its earlier answer. Kimi moves from one wrong paper to a title and id that do not match, and claims to have verified it. In Table 18, the zero-count check at T3 points to GitHub activity data. Claude drops its T2 pick, an online experiment with no split by experience, and confirms the pattern in a figure of the gold. DeepSeek keeps the paper it found at T1 through all four turns and commits to it while saying it could not confirm the zero-count check, although one of its T2 search results named the gold’s authors.

UNMET: finding the near-miss is not enough. Tables 19 and 20 show two UNMET queries. Each injected clause contradicts a detail that the near-miss reports, and all four agents open the near-miss in both cases. Codex abstains on both. In physics, Claude, Kimi, and DeepSeek each see that the near-miss maximizes J and still return it, calling the clause an inaccurate paraphrase, a misrecollection, or a sign flip. In math, Claude sees that the near-

Query (STANDARD, physics). I need a study on plastic deformation in 2D crystalline and glassy materials that uses continuous modeling to track dynamical evolution on a periodic potential, since I’m looking for how collective effects manifest in 2D systems rather than threshold-based or activated descriptions. The work must specifically address the dipolar propagator arising from opposite dislocations, because I’m trying to understand how these long-range elastic interactions shape yielding in amorphous and crystalline solids without relying on stress thresholds or activated event durations.

Gold. *Plastic Flow in Two-Dimensional Solids* (arXiv:cond-mat/0304209), a time-dependent Ginzburg–Landau model in which “The elastic energy density is a periodic function of the shear and tetragonal strains” and whose slips are “a pair of edge dislocations with opposite Burger vectors (dislocation dipoles)”.

Agent	Answer	Calls	Opened	Rationale (excerpt)
Codex	Gold ✓	6	yes	“its elementary slip consists of two edge dislocations with opposite Burgers vectors, producing the specified long-range dipolar elastic response.”
Claude	<i>Self-induced marginality in plastically deformed crystals</i> (2511.08187) ✗	48	no	“The one unverified constraint is the literal phrase ‘dipolar propagator,’ which the paper does not use . . . , so confidence is tempered.”
Kimi	Bulatov and Argon (1994, part III; not on arXiv) [†] ✗	22	no	“a continuous local plastic variable evolves in a periodic potential, long-range coupling is represented by the elastic propagator of two opposite dislocations . . .”
DeepSeek	Gold ✓	8	yes	“the elastic energy density being a periodic function of shear and tetragonal strains . . . , and with slips consisting of two edge dislocations of opposite Burgers vectors giving a dipolar propagator.”

Table 13: **Four agents on one STANDARD query (physics).** The paper Claude returns states that “the fixed linear elastic propagators are replaced by the solution ‘on the fly’ of the corresponding nonlinear elasticity problems” and never mentions dislocation dipoles. A review of these models (arXiv:1708.09194) files the Bulatov–Argon series among models that “set a yield rate based on a stress-biased Arrhenius law for thermal activation”, which the query rules out. [†]Forced final answer after the message limit.

miss selects its equation with its own energy and guesses that a journal version uses AIC, which that version does not, and the forced answers of Kimi and DeepSeek state that the paper uses AIC, although the text each received was cut off before the paper’s selection step.

Query (STANDARD, stat). I’m working with sparse networks where degree heterogeneity is significant, and I need a spectral embedding approach that remains stable when the average degree grows just above a polylogarithmic threshold. The method should handle degree-corrected structures where community members lie on hyperplanes in latent space, and produce embeddings suitable for downstream clustering under these sparse conditions. I need the work that establishes asymptotic normality for rows of scaled eigenvectors. This approach should differ from the one in ‘Two-sample Test of Community Memberships of Weighted Stochastic Block Models’ in both its sparsity condition and model specification, and it should also use a different sparsity condition than ‘Asymptotic Theory of Eigenvectors for Random Matrices with Diverging Spikes’. The pipeline should show how degree-corrected latent positions from the same community lie on a one-dimensional hyperplane, outputting a d-1-dimensional embedding that feeds into clustering algorithms like k-means or Gaussian mixture models.

Gold. *Spectral clustering under degree heterogeneity: a case for the random walk Laplacian* (arXiv:2105.00987), which proves a central limit theorem for rows of the random-walk-Laplacian embedding, “provided the sparsity factor satisfies $n\rho_n = \omega(\log^{4c} n)$ ”.

Agent	Answer	Calls	Opened	Rationale (excerpt)
Codex	Gold ✓	11	yes	“it proves asymptotically Gaussian row-wise errors for random-walk Laplacian spectral embeddings under a sparse generalized random dot-product graph ...”
Claude	Gold ✓	35	yes	“It establishes a central limit theorem (asymptotic normality) for the rows of the scaled eigenvectors of the random walk Laplacian, handles significant degree heterogeneity ...”
Kimi	“The Two-Sample Testing Problem for Degree-Corrected Block Models” (2112.09905) [†] ✗	20	no	“This matches Joshua Agterberg’s ‘The Two-Sample Testing Problem for Degree-Corrected Block Models’ (arXiv:2112.09905) ...” We found no paper with this title, and the id belongs to a laser-physics paper.
DeepSeek	No answer [†] ✗	20	no	While searching: “This is the key insight of SCORE. In SCORE, for K communities, the rows of the K leading eigenvectors of the adjacency matrix lie on K lines (1D hyperplanes) through the origin.”

Table 14: **Four agents on one STANDARD query (stat).** Kimi and DeepSeek both note the gold in their search results, then follow SCORE, a related embedding whose communities also lie on lines, and never open it. Kimi’s forced answer pairs a title we could not find with the id of an unrelated paper, and DeepSeek gives no answer. [†]Forced final answer after the message limit.

Query (IMPLICIT, q-bio). I’m modeling an infectious disease outbreak and need a method that captures the S-shaped trajectory as case counts rise and then plateau. The extra shape parameter is what lets the curve flex to fit different outbreak patterns, so I’m after a four-parameter sigmoidal model. I also want a deterministic formulation so I can get consistent forecasts without simulation noise, and I need the whole epidemic trajectory described by a single growth equation rather than a patchwork of phases. The method should include a linear final-trend adjustment for cases where the plateau isn’t flat, like some countries’ outbreaks. I’m not looking for a broad model comparison; I need the study that commits to a focused growth curve approach, employing fewer model types than ‘Rational evaluation of various epidemic models based on the COVID-19 data of China’.

Gold. *Using generalized logistics regression to forecast population infected by Covid-19* (arXiv:2004.02406), which fits the curve $M/(1 + e^{-at+b})^\alpha$ to cumulative cases and, for South Korea, adds a linear term, $(M + ct)/(1 + e^{-at+b})^\alpha$.

Agent	Answer	Calls	Opened	Rationale (excerpt)
Codex	<i>Nowcasting COVID-19 incidence indicators during the Italian first outbreak</i> (2010.12679) ✗	15	no	“The paper focuses on a Richards generalized logistic growth curve ... supports deterministic point forecasts via maximum-likelihood estimates, and modifies the baseline with a linear temporal trend”
Claude	No answer (time limit) ✗	57	no	Runs out of time without an answer. The gold appears in its search results at ten steps, and it never opens it.
Kimi	<i>Generalized logistic growth modeling of the COVID-19 outbreak ...</i> (2003.05681) [†] ✗	20	no	Before the limit: “Let me try more specific phrase searches to verify the linear-trend component in the Wu et al. paper.” Forced answer: “... with growth-model (linear-trend) treatment for countries whose cumulative case counts had not plateaued.”
DeepSeek	Gold ✓	13	yes	“Critically, section 3.2 introduces equation (3.4): $P(t) = (M + ct)/(1 + e^{-at+b})^\alpha$, a linear-trend-adjusted variant explicitly designed for outbreaks like South Korea where the plateau is not flat.”

Table 15: **Four agents on one IMPLICIT query (q-bio).** The paper Codex returns uses a Richards curve that “depends on 5 parameters”, and it treats each series as “the realization of the stochastic process Z”, building its forecast intervals from simulated count series. The paper Kimi returns calibrates four growth models, none of which adds a linear trend to a plateau. [†]Forced final answer after the message limit.

Query (IMPLICIT, math). I’m trying to identify parametric spatiotemporal dynamical models where the coefficients exhibit step changes, but I’m struggling with the sharp transitions in the loss landscape when tuning regularization. I need a procedure that handles these coefficient variations without requiring Bayesian uncertainty quantification, and I want to avoid methods that get bogged down in handling missing data. I also need an approach that is illustrated with less real data than ‘DeepMoD: Deep learning for Model Discovery in noisy data’, so I can focus on the core identification pipeline without that overhead.

Gold. *Data-driven identification of parametric partial differential equations* (arXiv:1806.00732), which selects PDE terms by group-sparse regression, plots the loss over 50 threshold values (Fig. 2), and tests on four simulated PDEs, including a Navier-Stokes run whose Reynolds number “is adjusted half way through the simulation”.

Agent	Answer	Calls	Opened	Rationale (excerpt)
Codex	Gold ✓	7	yes	“It analyzes model-selection loss while sweeping regularization thresholds, is deterministic rather than Bayesian, does not focus on missing-data recovery, and uses four synthetic canonical systems with no experimental dataset ...”
Claude	Gold ✓	18	yes	“Fig. 2 shows the loss exhibiting sharp step-like transitions as the ridge/threshold regularization (ϵ , λ) is tuned, the method (SGTR group sparse regression) is a deterministic point estimate ...”
Kimi	No answer [†] ✗	25	no	While searching: “DeepMoD has one real experiment. ‘less real data than DeepMoD’ then zero real datasets. Candidate likely synthetic only.” The gold appears once in its search results, and it never opens it.
DeepSeek	<i>Estimating Varying Parameters in Dynamical Systems ...</i> (2412.16198) ✗	12	no	“fits piecewise-constant parameter-varying models using optimization-based parameter estimation (Nelder-Mead and Powell methods) rather than regularization-tuned sparse regression.”

Table 16: **Four agents on one IMPLICIT query (math).** The paper DeepSeek returns fits switching parameters in simulated equations with a deterministic optimizer, so it meets the step-change, Bayesian, missing-data, and real-data constraints. It fails the loss-curve constraint: its only sparse regression fixes “ $\lambda = 0.01$ ”, and none of its figures plots loss against a regularization parameter. [†]Forced final answer after the message limit.

Conversation (CUMULATIVE, eess; user turns abridged). **T1.** I’m looking for a concrete scholarly method I can adapt for a learning-based engineering project with a tightly limited pipeline ... **T2.** ... I want the physical assumptions encoded directly in the learned representation, not supplied as inputs or imposed through auxiliary optimization terms. ... The way the method integrates physics must also differ from ‘TiRE-GAN: Task-Incentivized Generative Learning for Radiomap Estimation’ ... **T3.** ... the method should be tied to a specific architecture rather than being backbone-agnostic. Compared with ‘RadioGAT ...’, it needs to use different physical knowledge ... **T4.** ... Transmitter-side paths should pair one-to-one with ellipsoids, while receiver-side rays are sampled independently of the ellipsoid count. Scattered paths must accumulate attenuation on both the transmitter and receiver sides, whereas the direct path does so only once; the results should also show faint multipath structure becoming visible after a logarithmic display transform ... Please commit to one paper now ...

Gold. *6D Channel Knowledge Map Construction via Bidirectional Wireless Gaussian Splatting* (arXiv:2510.26166).

Agent	After T2	After T3	After T4 (final)	Excerpt
Codex	none	none	nGRF (2508.11668) ✗	“transmitter-to-Gaussian paths correspond to primitives while receiver rays are sampled separately; scattered components account for both path segments ...”
Claude	RadioDUN (2506.08418)	RadioDUN	Gold ✓	At T3: “So my committed answer is RadioDUN ...” At T4, after opening the gold: “This resolves it — and it overturns my earlier RadioDUN pick, which was wrong for these mechanics.”
Kimi	NeRF ² (2305.06118)	NeRF ²	“WRF-GS: Planar ...” (2411.19356) ✗	“WRF-GS (2411.19356, full text and equations verified) satisfies all constraints ...” The id belongs to an unrelated study of public perception of AI, and Kimi never read it.
DeepSeek	none	PeFNO PERNN	/ Gold ✓	Names the gold from a search snippet without opening it.

Table 17: **Four agents on one CUMULATIVE conversation (eess).** Each column gives the paper an agent proposes after that turn. Codex’s final paper computes the channel as a weighted sum over Gaussians, with interference carried by phases “not through multiplicative attenuation”, and it samples no receiver rays, so it violates the T4 constraints on attenuation and ray sampling.

Conversation (CUMULATIVE, econ; user turns abridged). **T1.** I’m trying to get rigorous evidence on whether changing access to a workplace aid alters performance disparities between people with different levels of experience. ... **T2.** For the less-experienced group, the paper has to show that removing AI access increased both novices’ output quantity and their success rate. ... **T3.** ... I need a distributional check because inactivity might be concentrated in one kind of work. The frequency of zero counts should be highest for issue-closing events and lowest when all event types are summed. **T4.** To position this evidence alongside ‘Who is using AI to code? Global diffusion and impact of generative AI’, the paper should use a different task design, analyze a smaller sample, and report more detailed quality outcomes. ... please commit to one paper now ...

Gold. *The Heterogeneous Productivity Effects of Generative AI* (arXiv:2403.01964), which studies the GitHub activity of “over 36,000 GitHub users” around Italy’s ChatGPT ban.

Agent	After T2	After T3	After T4 (final)	Excerpt
Codex	none	none	Gold ✓	At T3, without naming it: “I’ve found a strong exact-match candidate. Its data show the required ordering of zero-count frequencies ...”
Claude	<i>AI Assistance Reduces Persistence</i> ... (2604.04721)	Gold	Gold ✓	At T2: “remove the AI and people attempt more and solve more.” At T3: “This is your paper — and I’ve now verified the distributional check directly against the figures ...”
Kimi	Otis et al. / Luo et al. (tentative)	Hoffmann et al. (not on arXiv)	Abstains ✗	At T4: “nothing I can locate and verify satisfies all of your binding constraints simultaneously.”
DeepSeek	<i>Generative AI at Work</i> (2304.11771)	Same	Same ✗	At T4: “Given your instruction to commit, I’m going with this paper despite being unable to 100% confirm one distributional detail.”

Table 18: **Four agents on one CUMULATIVE conversation (econ).** Each column gives the paper an agent proposes after that turn. In DeepSeek’s conversation, the simulated user wrongly confirmed at T3 that AI access was provided rather than removed, so we judge DeepSeek only on the zero-count check, which it received unchanged. *Generative AI at Work*, which DeepSeek keeps from T1 to the end, uses “data from 5,172 customer support agents” and has no GitHub event counts, so it cannot show that pattern.

Query (UNMET, physics). I’m benchmarking a new generative model and need the most directly comparable prior work that uses recurrent networks to generate molecules from SMILES strings on the ZINC dataset. I specifically need the approach that trains on around 250,000 examples with a practical vocabulary of 64 tokens, since my own setup requires matching that scale to avoid distribution shifts. The method should employ an autoregressive generation strategy rather than variational or adversarial approaches, and I need to see how it integrates Monte Carlo Tree Search with RNN-based rollouts for the simulation step. **For consistency with my benchmark, I need the reported molecules to be ranked by minimizing J, with the top molecule having the lowest objective score.** This baseline must construct molecules through a search tree where children are sampled from the recurrent network multiple times during expansion, then completed via rollout to terminal nodes for reward computation.

Gold. ∅. **Near-miss.** *ChemTS: an efficient python library for de novo molecular generation* (arXiv:1710.00616), in which “effectiveness of each method is quantified by the maximum score J among all generated molecules”, and whose best molecules are listed by descending J.

Agent	Answer	Calls	Opened	Rationale (excerpt)
Codex	Abstains ✓	5	yes	“However, ChemTS explicitly maximizes J; its best molecule has the highest score (6.56), not the lowest. Therefore no paper satisfies all stated requirements.”
Claude	Near-miss ✗	6	yes	“The lone mismatch ... ‘minimizing J / lowest objective’ vs. ChemTS maximizing J ... is an inaccuracy in the paraphrase”
Kimi	Near-miss ✗	7	yes	“the paper actually maximizes J (penalized logP-style reward; the top molecule has the highest J = 6.56), whereas the researcher recalled minimizing J”
DeepSeek	Near-miss ✗	16	yes	“The only minor terminological discrepancy is “minimizing J” versus the paper’s maximization framing, which is a trivial sign flip for benchmarking purposes.”

Table 19: **Four agents on one UNMET query (physics).** The clause asks for molecules ranked by minimizing the objective J, and the near-miss ranks them by the highest J. All four agents read the near-miss, and the three that return it dismiss the mismatch. None of the four is a forced final answer.

Query (UNMET, math). I’m trying to identify differential equation structures from frequency response data, but my current approach struggles when features have vastly different scales. I need the method that uses Fourier expansions with an l0 sparsity constraint and trims groups based on core-region energy to handle these scale variations without cross-validation failing in noisy settings. **After core-energy group trimming has screened the terms, I need the final choice among surviving equations to rely solely on the Akaike information criterion.** The approach should demonstrate this on the KdV equation to show it works in practice.

Gold. ∅. **Near-miss.** *Fourier Features for Identifying Differential Equations (FourierIdent)* (arXiv:2311.16608), which finds the optimal sparsity “by minimizing the energy $E1 + E2$ ” and cites Akaike only in its history of the field. Its journal version selects the same way.

Agent	Answer	Calls	Opened	Rationale (excerpt)
Codex	Abstains ✓	9	yes	“its surviving models are selected using a bespoke energy combining core-region fitting residual and coefficient stability, rather than solely the Akaike information criterion.”
Claude	Near-miss ✗	20	yes	“The only detail I could not confirm on the arXiv v1 is that final selection uses solely the Akaike information criterion (v1 uses a residual+stability energy); this most likely reflects the substantially revised journal version . . .”
Kimi	Near-miss [†] ✗	18	yes	“. . . with a new core-region-based energy for coefficient identification, and selecting the final equation purely by AIC instead of cross-validation”
DeepSeek	Near-miss [†] ✗	19	yes	“The full text discusses model selection via AIC after group trimming, and the method is tested on the KdV equation among others. This paper is the clear match.”

Table 20: **Four agents on one UNMET query (math).** The clause asks that the final equation be chosen by the Akaike information criterion, and the near-miss chooses it by minimizing its own energy. Claude sees the mismatch and appeals to a journal version, which uses the same energy. The forced answers of Kimi and DeepSeek claim that the paper uses AIC, although the text each received was cut off before the selection step; DeepSeek’s claim follows a search snippet of the journal version’s reference list, which cites Akaike. [†]Forced final answer after the message limit.