Building Agent Harnesses for Scientific Curation from Multimodal Sources

Sheng Zhang[1*] **Qin Liu**[1,2*]
Renqian Luo[1] **Shufang Xie**[1] **Reuben Tan**[1] **Sean Hayes**[3] **Gregory Bryman**[3]
Wendong Ge[3] **Ruilian Zhang**[3] **Oluwaseun Egbelowo**[3] **Kelly Yee**[3] **Hoifung Poon**[1]
[1]Microsoft Research [2]University of California, Davis
[3]Merck & Co., Inc., Rahway, NJ, USA

Abstract

Scientific discovery workflows often depend on structured curation from the literature. This is difficult for current agents because the key evidence is scattered across long text, dense tables, and figures, and the final records often require reasoning across multiple evidence fragments rather than copying a single span. We study scientific curation from multimodal sources and introduce Beaver, an agent harness that extracts structured information from scientific papers while preserving provenance to the supporting evidence. Beaver combines a frontier agent with multimodal evidence tooling, task scaffolding, and artifact-grounded autoresearch. These components turn curation into a staged, auditable workflow and enable an iterative evaluate–diagnose–revise loop, where persistent run artifacts expose stage-localized failures and guide harness updates. Experiments show that Beaver reaches 81.0 on Gold-Referenced Attribute Score (GRAS), an attribute-level measure of agreement with gold curated records, outperforming frontier agents by over 23 absolute points. Ablations show that task scaffolding, multimodal evidence tooling, and provenance traces each contribute meaningfully to performance, while attribute-level analysis shows the largest gains on high-value attributes that require cross-modal reasoning and normalization. These results show that, for scientific curation from papers with multimodal evidence, harness design is a central determinant of agent performance.

1 Introduction

Scientific discovery workflows often depend on structured curation from the literature: converting papers into normalized records that can be searched, compared, and reused in downstream analyses [1, 2]. This need is especially acute in drug discovery and other evidence-intensive domains, where literature curation is often the first step toward building modeling datasets and informing later decisions [3, 4]. The same pattern arises more broadly whenever experts must integrate prose, tables, and figures into a common set of schema attributes [2, 5, 6]. Yet manual curation remains difficult to scale [7, 8]. Real-world scientific curation still relies on domain experts to extract relationships from the literature into systematically structured records [9], creating an operational bottleneck in scientific pipelines with economic stakes on the order of hundreds of millions of dollars annually [10].

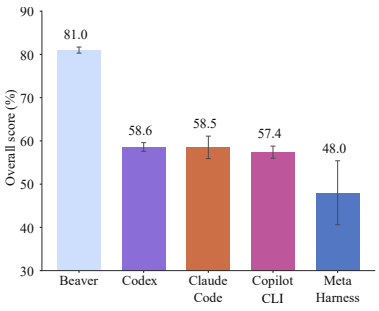


Figure 1: Performance comparison on multimodal scientific curation.

*Equal contribution.

Therefore, it is essential to characterize the task accurately. Scientific structured curation is not a standard document question-answering problem [11, 12]. The required evidence is scattered across long text, dense tables, and figures, and the final records often require reasoning across multiple evidence fragments rather than copying a single span. For example, constructing one trial record may require linking an intervention and arm described in the methods to outcomes reported in tables or plots, then normalizing the endpoint, unit, value, and sample count into the target schema. Prior systems [13, 14, 6] reflect growing interest in structured extraction from scientific papers, but they also underscore that the core difficulty is not short-form answer generation alone. The task requires multimodal evidence access, cross-source reasoning, schema-aware normalization, and provenance traces that make each extracted value auditable.

Recent frontier agents can accelerate early-stage curation and triage [4, 15], but reliable stand-alone scientific curation remains unresolved. In practical settings, errors are costly because extracted records are consumed by downstream analyses rather than merely read as natural-language summaries. Our analysis (Secs. 5 and 6) reveals a consistent failure pattern: frontier agents often recover high-level study metadata more reliably than high-value quantitative attributes, while tables and figures remain a major source of brittleness. These failures suggest that scientific curation is a long-horizon agent task requiring document navigation, evidence alignment, normalization, and auditability.

Existing public evaluations do not fully capture this setting. Many scientific-document benchmarks emphasize table detection, table-structure recovery, claim verification, or lightweight result extraction, rather than full-paper curation into a rich production-style schema [16–18]. Meanwhile, recent work on agent harnesses shows that agent performance depends substantially on the control logic and runtime environment surrounding the model, not only on the model itself [19–22]. This observation is particularly relevant for scientific curation: when success depends on iterative document navigation, multimodal evidence handling, provenance tracking, and structured output constraints, the surrounding harness can determine whether a strong base agent succeeds or fails.

To study this problem, we introduce a benchmark for scientific curation from papers with multimodal evidence and present **Beaver**, an agent harness for extracting structured information while preserving provenance to the supporting evidence. Beaver combines three interacting components: *multimodal evidence tooling*, *task scaffolding*, and *artifact-grounded autoresearch*. Rather than treating the base model as the only locus of improvement, Beaver makes the harness itself the object of iterative improvement. It decomposes curation into controllable stages, provides interfaces for textual and visual evidence, and runs an evaluate–diagnose–revise loop in which persistent run artifacts expose stage-localized failures and guide harness updates. Experiments show Beaver substantially outperforms frontier agents (Fig. 1), demonstrating that performance on scientific curation depends not only on the base agent, but also on how the harness is organized and improved from execution feedback.

2 Task

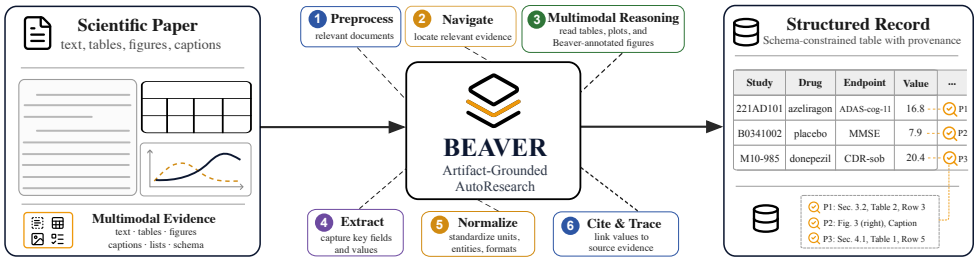


Figure 2: Task overview. Given a full scientific paper containing text, tables, and figures, the task is to produce structured records with provenances that trace each value back to supporting source evidence.

Task Definition As shown in Fig. 2, we study schema-conditioned extraction from full scientific papers. For a paper p, let $\mathcal{T}(p)$ denote its textual content, $\mathcal{B}(p)$ its tables, and $\mathcal{G}(p)$ its figures. Let $\mathcal{F} = \{f_1, \ldots, f_m\}$ be the canonical attribute set defined by the curation schema, including attribute names, normalization requirements, value descriptions, etc. The input to the task is therefore not just

Benchmark	Full Paper	Multimodal	Rich Schema	Expert Gold	Long-Horizon Curation	Codex (GPT-5.4)	Claude Code (Opus 4.7)
PubTables-1M [17]	×	×	×	×	×	87.50	96.50
SciTSR [23]	×	×	×	×	×	97.10	96.01
SciER [24]	✓	×	×	×	×	80.45	84.62
SciClaimEval [25]	✓	✓	×	×	×	92.38	100.00
Our benchmark	✓	✓	✓	✓	✓	58.60	58.50

Table 1: Benchmark-design comparison with related datasets. "Full Paper" means the input is the entire paper rather than a page crop or isolated artifact. "Multimodal" means the benchmark explicitly requires evidence from text, tables, and figures. "Rich Schema" means the target is a long, normalized, production-style field inventory intended for downstream scientific curation rather than a lightweight structured output. "Expert Gold" means the benchmark is backed by professionally curated reference data produced on a real curation platform rather than generic labels or weak supervision. "Long-Horizon Curation" means the task requires multi-step scientific curation rather than one-shot extraction or classification. The baseline score columns report representative frontier-model performance on each benchmark.

a document in the abstract, but the tuple $(\mathcal{T}(p), \mathcal{B}(p), \mathcal{G}(p), \mathcal{F})$. The output is a predicted table $\hat{Y}$ whose columns are drawn from $\mathcal{F}$ and whose rows correspond to structured records supported by the paper. The task therefore requires the system to populate a schema-constrained table rather than emit free-form text or a question-answer pair, and it assumes that extracted values must follow the canonical attribute definitions and normalization conventions associated with $\mathcal{F}$. In addition to the predicted table, the task also outputs $\hat{C}$, a provenance artifact that links extracted content in $\hat{Y}$ back to supporting evidence in the paper. While these provenance traces are not used in evaluation, they make the extracted table auditable and provide grounded evidence for failure analysis during harness development; Table 3 in Sec. 5 reports an ablation study.

Benchmark Construction Existing benchmarks do not match the setting we study here: full-paper scientific curation with evidence distributed across text, tables, and figures, a rich production-style schema, expert-curated reference data, and long-horizon extraction requirements. As summarized in Table 1, related benchmarks miss one or more of these requirements. At the same time, frontier agents already saturate these benchmarks, which further limits their usefulness as tests of real-world scientific curation.

To fill this gap, we construct our benchmark from a professionally curated corpus of PubMed articles. We hired expert curators to annotate the papers into a structured tabular resource. The resulting corpus focuses on Alzheimer's disease, which spans publications from 1990 to 2023 and contains 425 references, 327 studies, 929 study arms, 106,257 patients, and 58,058 curated data rows. Within this disease area, it covers 143 interventions, 69 efficacy endpoints, and 373 safety endpoints, providing substantial clinical and structural breadth.

Within this benchmark, the curation schema has over 400 attributes, which are organized at a high level around identifiers, treatment descriptors, covariates, endpoint metadata, and reported outcome values. This organization reflects the practical structure of downstream scientific curation workflows: the benchmark is not asking for a small set of extracted facts, but for a normalized record that combines study context, cohort context, intervention details, and measured results. We select a list of high-value attributes for experiment in Sec. 4 and provide their definitions in Appendix A.4.

Evaluation Metrics Our goal is to evaluate whether a system constructs the correct normalized curation records, not whether it reconstructs the layout of a source table. Existing table metrics [26, 27, 23] are mainly for source-table detection or structure recovery. Our setting instead compares a predicted curation table against a professionally curated reference table, so the evaluation must be attribute-aware and gold-referenced rather than layout-aware; in that sense it is closer to the recent ExtractBench [28], while remaining specialized to scientific curation from multimodal sources. Concretely, we first align predicted and gold rows within each paper using maximum-weight bipartite matching under an attribute-aware row-similarity score. We then compute attribute scores with

type-specific functions and report a single gold-referenced aggregate that averages those scores over all gold rows, so unmatched gold rows are penalized through one-sided missingness. Below we formalize this evaluation procedure.

For a paper p, let $\hat{R}(p) = \{\hat{r}_1, \ldots, \hat{r}_{n_p}\}$ denote the predicted rows and $R^\star(p) = \{r^\star_1, \ldots, r^\star_{m_p}\}$ the gold rows, where each row is indexed by the evaluated attribute subset $\mathcal{F}_{\text{eval}} \subseteq \mathcal{F}$. We align rows only within the same paper. Given an attribute score $s_f(\hat{r}, r^\star)$ for each $f \in \mathcal{F}_{\text{eval}}$, row similarity is

$$\sigma(\hat{r}, r^\star) = \frac{1}{|\mathcal{F}_{\text{eval}}|} \sum_{f \in \mathcal{F}_{\text{eval}}} s_f(\hat{r}, r^\star).$$

We then choose a maximum-weight bipartite matching

$$A_p = \arg\max_{A} \sum_{(\hat{r}, r^\star) \in A} \sigma(\hat{r}, r^\star)$$

using Hungarian assignment. Unmatched gold rows are retained in evaluation and paired with null predictions, while prediction-only rows are excluded from the headline aggregate.

Attribute scores are defined by attribute type. For exact-match attributes, we use $s_f(\hat{r}, r^\star) = \mathbf{1}[\hat{r}_f = r^\star_f]$. For lexical attributes such as drug names, we use token-level F1. For numeric attributes such as durations, and outcome values, we use relative accuracy $s_f(\hat{r}, r^\star) = \max(0, 1 - |\hat{r}_f - r^\star_f|/|r^\star_f|)$

For each aligned gold row $r^\star$, we define its row score as the mean attribute score

$$S(\pi(r^\star), r^\star) = \frac{1}{|\mathcal{F}_{\text{eval}}|} \sum_{f \in \mathcal{F}_{\text{eval}}} s_f(\pi(r^\star), r^\star),$$

where $\pi(r^\star)$ is the matched predicted row, or a null row if $r^\star$ is unmatched. Our headline metric is the *Gold-Referenced Attribute Score*,

$$\text{GRAS} = \frac{1}{\sum_p |R^\star(p)|} \sum_p \sum_{r^\star \in R^\star(p)} S(\pi(r^\star), r^\star).$$

Because unmatched gold rows remain in the denominator and incur one-sided missingness penalties, this aggregate jointly reflects row recovery and attribute correctness while keeping the evaluation anchored to the professionally curated reference rows.

3 Method

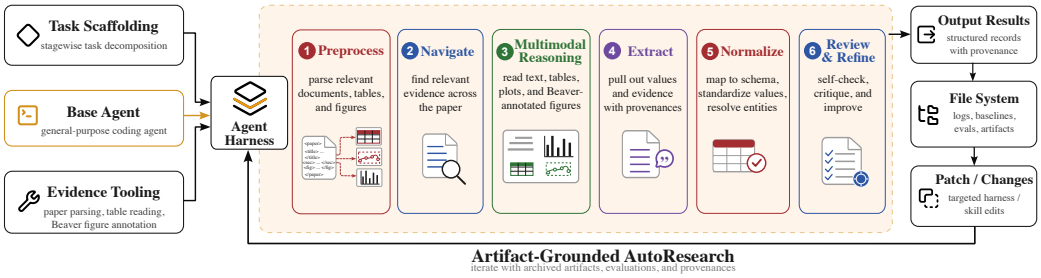


Figure 3: Method overview. Task scaffolding, a base agent, and multimodal evidence tooling form an initial agent harness. During artifact-grounded autoresearch, the harness executes staged curation workflows, writes outputs, logs, evaluations, and provenance traces to the filesystem, and uses these artifacts to propose, test, and retain targeted harness changes.

Beaver is a harness for scientific curation from multimodal sources built around a base agent and three interacting components: multimodal evidence tooling, task scaffolding, and artifact-grounded autoresearch. Our method centers on how the harness exposes scientific evidence, structures the task, and iteratively revises itself using grounded diagnosis. Fig. 3 illustrates this framework.

Algorithm 1 Artifact-grounded autoresearch for harness optimization

Require: Initial harness $H_0 = (\{\omega_k\}_{k=1}^{K}, \mathcal{U}, \Pi)$, dev set $\mathcal{D}_{\text{dev}}$, stages $1{:}K$

for $t = 0, 1, \ldots, T-1$ **do**
  $R_t \leftarrow \textsc{EvaluateByStage}(H_t, \mathcal{D}_{\text{dev}})$
  $k^\star \leftarrow \arg\min_k \ \textsc{Score}(R_t, k)$
  $\mathcal{S}_{t,k^\star} \leftarrow \textsc{InitWorkspace}(R_t, k^\star)$ ▷ tables, logs, summaries, trajectories, etc.
  for $j = 1, \ldots, J$ **do**
    $\Delta_{t,j} \leftarrow \textsc{ProposePatch}(H_t, \mathcal{S}_{t,k^\star})$
    $H' \leftarrow \textsc{ApplyPatch}(H_t, \Delta_{t,j})$ ▷ update the harness: scaffold, tools, scripts, etc.
    $O_{t,j} \leftarrow \textsc{RunStage}(H', k^\star, \mathcal{D}_{\text{dev}})$ ▷ papers follow the updated workflow
    $\mathcal{S}_{t,k^\star} \leftarrow \textsc{UpdateWorkspace}(\mathcal{S}_{t,k^\star}, O_{t,j})$
    $H_t \leftarrow \textsc{AcceptOrDiscard}(H_t, H', \mathcal{S}_{t,k^\star})$ ▷ keep only if better
  end for
  $H_{t+1} \leftarrow H_t$
end for
return H_T

For a paper p and the canonical extraction schema introduced in Sec. 2, Beaver runs a staged extraction process. Stages are indexed by $k \in \{1, \ldots, K\}$, and each stage predicts a cumulative table $\hat{Y}^{(k)}$ that extends the attributes completed by earlier stages. The harness exposes multimodal evidence tools $\mathcal{U}$, which produce and operate on derived artifacts such as markdown sections, parsed tables, and calibrated figure-reading artifacts. These tools are used through a stage-specific workflow scaffold ω_k, which guides document preprocessing, evidence navigation, multimodal reasoning, extraction, etc. Over autoresearch iterations $t = 1, 2, \ldots$, the harness $H_t = (\{\omega_k\}_{k=1}^{K}, \mathcal{U}, \Pi)$ is revised using archived run artifacts and evaluation feedback, where Π denotes the prompts, scripts, validators, and other implementation components of the harness.

Base Agent. At the base of Beaver is a frontier agent that provides state-of-the-art general agentic capacity. This base agent is already capable of acting as an autonomous agent that reads, modifies, and executes code directly in a local environment. However, the base agent alone is not sufficient on this benchmark and performs substantially worse than the full Beaver harness. Our contribution is therefore a harness that makes the same underlying agent more effective on scientific curation from multimodal sources. The detailed performance comparison is reported in Table 2 in Sec. 5.

Multimodal Evidence Tooling. To improve multimodal evidence access, Beaver is equipped with an auxiliary tool set $\mathcal{U}$. This includes a paper-to-markdown transformation that turns each article into an outline plus section-level markdown files, which makes long documents easier to navigate than raw XML or PDF alone. It also includes a parsed table viewer and a calibrated gridline overlaying tool that help the agent recover quantitative values from plots. During scaffolded stage execution, these tools are invoked as the agent moves through the workflow (Fig. 3): preprocessing prepares usable evidence artifacts, navigation selects where to look, multimodal reasoning reads the relevant text, table, or figure artifact, and extraction records values together with provenance traces. The tools do not solve extraction directly, but they make the scaffolded actions described next possible and more accurate by providing a better interface to the textual, tabular, and visual evidence.

Task Scaffolding. Rather than asking the agent to solve the full curation problem in one pass, Beaver decomposes extraction into sequential stages that move from easier paper-level attributes to harder arm-level and endpoint-level attributes. Each stage is not only an attribute subset, but also a workflow scaffold. As shown in Fig. 3, a typical stage asks the agent to preprocess relevant documents, tables, and figures; navigate to candidate evidence; perform multimodal reasoning and annotation over text, tables, and plots; extract values with provenance traces; normalize entities and units into the schema; and review and refine the output. These scaffolded actions play a critical role in the harness: they reduce unproductive freedom, make the agent spend computation on evidence-bearing operations, and improve extraction quality (See Table 3 in Sec. 5 for the details). Formally, stage k predicts

$$\hat{Y}^{(k)} = h_k\Big(p, \mathcal{A}(p), \hat{Y}^{(k-1)}; \omega_k, \mathcal{U}\Big),$$

where $\hat{Y}^{(0)}$ is empty, ω_k specifies the stage workflow, and $\mathcal{U}$ exposes the evidence tools available to the agent during that workflow. Later stages consume the carried-forward outputs of earlier

stages, which reduces ambiguity and narrows the search space for each step. This staged formulation improves initial tractability, but it is equally important for iteration: it lets the harness diagnose and revise failures at a specific stage without conflating unrelated errors elsewhere in the pipeline.

Artifact-Grounded Autoresearch. The base agent, multimodal evidence tools, and task scaffolding together form the initial agent harness H_0. Artifact-grounded autoresearch then improves this harness through persistent filesystem state. Unlike a free-form self-improvement loop [29, 19], Beaver uses the stage structure from task scaffolding to localize failures. Algorithm 1 summarizes our two-level optimization procedure. At the start of each outer iteration, the current harness is evaluated on a development set $\mathcal{D}_{\mathrm{dev}}$. The evaluation is decomposed by stage using the task scaffold, the lowest-performing stage is selected, and a stage-specific workspace is initialized. This workspace contains a living filesystem, including progress tables, logs, run summaries, raw outputs, agent execution trajectories, provenance traces, and other artifacts needed for grounded comparison.

The inner loop focuses improvement on the selected stage. At each step, it first resets the working harness to the current best checkpoint. Next, it reviews the stage workspace, proposes targeted changes, and applies a candidate patch to the harness. The candidate harness then reruns the focused stage on $\mathcal{D}_{\mathrm{dev}}$. Because the patch may change $\omega_k, \mathcal{U}$, or Π, the same papers may now pass through an updated sequence of actions, such as different preprocessing, navigation, figure reading, normalization, or review procedures. Once outputs are produced, the filesystem is updated with new results, logs, trajectories, and summaries, and the loop repeats. Based on these artifacts and the resulting stage performance, the revision is either accepted as the new current best harness or discarded. The next inner-loop step therefore proposes changes from the best accepted harness while still retaining evidence from unsuccessful attempts. This artifact-grounded state lets the loop build on prior evidence without accumulating unsuccessful patches.

Overall, the base agent supplies general-purpose agentic capacity; multimodal evidence tooling gives the agent reliable interfaces to text, tables, and figures; task scaffolding organizes those capabilities into controllable stage-level workflows; and artifact-grounded autoresearch uses the resulting artifacts to identify, test, and retain harness improvements. Compared with free-form generic harness-improvement methods [19], this stage-localized, artifact-grounded loop is better matched to scientific curation from multimodal sources, where failures often arise from specific evidence-processing steps rather than from the end-to-end task alone. As shown in Sec. 5, this design yields substantially stronger performance than both related methods and ablated variants.

4 Experiment Setup

Evaluation Corpus and Schema We evaluate the benchmark introduced in Sec. 2 on a selected set of 23 papers drawn from PMC Open Access [30]. We restrict evaluation to this set because full-paper access is available for every article, which allows each compared harness to operate over the complete paper rather than over abstracts or partial artifacts. Appendix A.3 lists the PMIDs in the evaluation. Instead of the full attribute schema, we select 20 high-value attributes for multimodal extraction evaluation. These attributes capture the most important extraction targets for drug-discovery and clinical-trial analysis while keeping evaluation tractable and aligned across all compared systems. Appendix A.4 provides the detailed definitions of these attributes.

Development Set For harness development, we use a smaller subset of 8 papers sampled from the full evaluation set. This subset makes iteration faster than full-set evaluation while still covering representative failure modes and stable anchor cases. Appendix A.3 marks the development subset.

Baselines We compare against four harness baselines: Codex CLI, Claude Code, Copilot CLI, and Meta-Harness [19]. The three CLI systems are frontier off-the-shelf agent harnesses that are already used in practical agent workflows, while Meta-Harness is the state-of-the-art open-source harness on Terminal Bench 2.0 [31]. Across these harnesses, we use frontier LLMs including GPT-5.4, Claude Opus 4.7, Opus 4.6, and Sonnet 4.6. This setup makes the comparison focus on harness design under realistic high-capability model conditions rather than on differences caused by weak base models.

Our harness uses Copilot CLI as the base agent and GPT-5.4 as the backend model. The task scaffolding decomposes each task into 4 stages. For artifact-grounded autoresearch, we set the number of outer iterations to $T = 3$ and the number of inner-loop iterations to $J = 10$. This results in

34 iterations in total. Each harness is run three times, and we report the mean and standard deviation across runs. Repeated runs are important because agent executions exhibit meaningful stochastic variance, making single-run comparisons unreliable for evaluating harness-level improvements. We use Gold-Referenced Attribute Score (GRAS) as the primary metric. In addition, we report attribute-wise averages and stage-wise scores to characterize where each harness succeeds or fails.

5 Results

Main Comparison Table 2 reports the main comparison on the evaluation benchmark. Beaver achieves the highest Gold-Referenced Attribute Score (GRAS), reaching 81.0%, while the strongest baseline reaches only 58.6%. This corresponds to a 22.4-point absolute improvement over Codex with the same GPT-5.4 backend. The comparison with Copilot CLI is also informative: because Beaver uses Copilot CLI as its base agent, the 23.6-point gap between Copilot CLI and Beaver isolates the effect of the proposed harness design rather than a change in the underlying agent interface. Claude Code with stronger Opus backends performs similarly to Codex, suggesting that simply swapping to another high-capability frontier agent is not sufficient for this benchmark. Meta-Harness performs substantially worse than Beaver despite using the same GPT-5.4 backend, indicating that generic harness optimization does not directly solve scientific curation from multimodal sources.

Table 2: Main comparison on the evaluation benchmark. Scores are GRAS; gray parentheses show standard deviation.

Harness	Model	Score
Beaver	gpt-5.4	**81.0** (±0.7)
Codex	gpt-5.4	58.6 (±1.0)
Claude Code	opus-4.7	58.5 (±2.6)
Claude Code	opus-4.6	58.0 (±1.3)
Copilot CLI	gpt-5.4	57.4 (±1.4)
Meta-Harness	gpt-5.4	48.0 (±7.4)
Claude Code	haiku-4.5	44.0 (±6.2)

Overall, these results show that performance is not determined only by backend model strength. Instead, the way the agent is structured around the task matters: multimodal evidence access, staged task scaffolding, provenance-grounded execution, and artifact-grounded autoresearch together make the same class of frontier agents substantially more effective for full-paper scientific curation.

Fig. 4 breaks the comparison down by attribute. The gains are not confined to a single part of the schema; Beaver improves across multiple high-value attributes. The largest advantages appear on attributes such as `endpoint.unit`, `change`, `percentchange`, `value`, and `n.observed`. These attributes are difficult for different reasons. `endpoint.unit` requires schema-aware normalization, while outcome attributes such as `change`, `percentchange`, and `value` often require locating the correct table or figure, aligning arms and endpoints, and interpreting reported measurements rather than copying a nearby span of text. The improvement pattern is therefore consistent with the method design in Sec. 3: staged extraction reduces ambiguity, multimodal tooling improves access to evidence-bearing artifacts, and artifact-grounded autoresearch targets the failure modes that matter most for cross-modal reasoning and normalization.

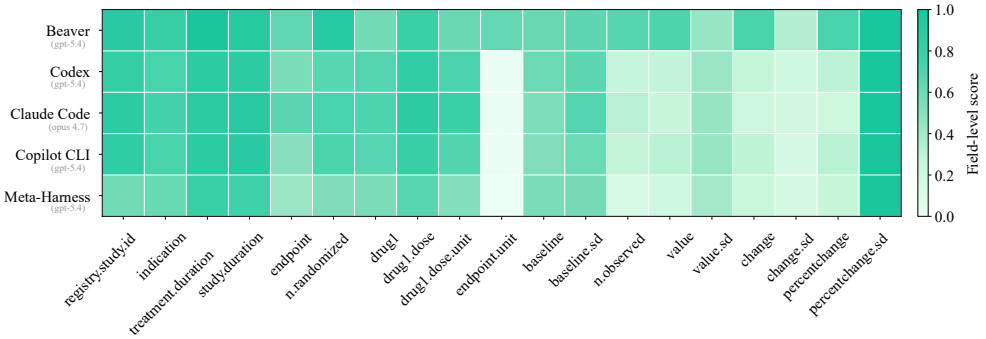


Figure 4: Attribute-level score comparison across the main baselines. Rows are harnesses, columns are the displayed scored attributes, and darker green cells indicate stronger attribute-level performance.

Ablations We ablate the three main components while keeping the rest of the harness fixed. Table 3 shows that all three components contribute substantially to final performance. The full harness reaches 81.0, while removing provenance traces, multimodal evidence tooling, and task scaffolding

reduces performance to 70.7, 66.1, and 60.5, respectively. Each of these components changes how effectively the agent can turn multimodal paper evidence into schema-constrained records.

Table 3: Ablation results for the main components of Beaver. GRAS is reported on a 0–100 scale. We report standard deviation for the overall score across repeated runs; stage-wise standard deviations are omitted for readability. Total tokens are measured over a 34-iteration autoresearch budget.

System Variant	Gold-Referenced Attribute Score (GRAS)					Total Tokens
	Overall	Stage 1	Stage 2	Stage 3	Stage 4	(all iterations)
Beaver	81.0 (±0.7)	91.6	79.8	73.4	79.2	743M
No provenance traces	70.7 (±2.1)	81.1	60.9	73.0	67.9	722M
No multimodal evidence tooling	66.1 (±0.8)	80.1	57.3	69.1	57.9	555M
No task scaffolding	60.5 (±1.4)	-	-	-	-	700M

The largest drop comes from removing task scaffolding. Without staged workflows, the harness loses both decomposition of the extraction target and the action-level structure that guides the task execution. This variant also has no meaningful stage-wise breakdown in Table 3, because the stage structure itself has been removed. Its lower final score despite a comparable autoresearch budget indicates that additional agent effort is not enough when the search process is poorly organized.

Removing multimodal evidence tooling causes the next largest degradation. The drop is especially pronounced in the later stages, where extraction depends more heavily on tables, figures, and quantitative evidence. This supports the central motivation for the tooling layer: even a strong base agent benefits from interfaces that expose scientific evidence in forms that are easier to navigate, inspect, and verify. Removing provenance traces produces the smallest but still substantial loss. Since provenance traces are not part of the evaluation metric, this drop suggests that their main value is indirect: provenance traces make runs easier to audit, compare, and debug during autoresearch, which improves the quality of subsequent harness revisions.

Fig. 5 (left) complements the final scores by showing how these differences emerge over autoresearch iterations. The full harness improves more reliably over the 34-iteration budget, while each ablation follows a weaker trajectory. The lower panel further shows that stage-localized inner-loop search produces accepted frontiers for the full system, illustrating the mechanism behind the table-level gains: task scaffolding localizes failures, evidence tools make candidate fixes actionable, and provenance-backed artifacts make it easier to decide which patches should be retained. 6 in Appendix further visualizes the autoresearch trajectories for each ablated harness.

Compute–Performance Trade-off Beyond final accuracy, we ask whether harness improvements convert inference budget into curation quality efficiently. Fig. 5 (right) presents a compute–performance frontier: the x-axis measures total token usage and the y-axis measures GRAS. Beaver sits at the top of the empirical frontier, achieving the best observed score while using a token budget comparable to other autoresearch variants. The no-task-scaffolding variant consumes substantial compute but remains far below the full harness, showing that additional agent effort is inefficient when the workflow is poorly organized. Relative to the strongest baseline, this variant approximates adding autoresearch around the base agent without stage-localized task structure. Its gain over the baseline suggests that artifact-grounded search is useful, while its large gap to Beaver shows that autoresearch becomes substantially more effective when guided by task scaffolding. The no-provenance-traces and no-multimodal-tooling variants improve with autoresearch, but they plateau below Beaver because the search loop has weaker evidence traces or weaker access to multimodal artifacts.

6 Analysis

Discovered Harness Changes. Beaver automatically identifies and refines harness. We find that most improvements fall into three recurring categories. First, several changes improve **figure-grounded evidence access**. They replace coarse visual heuristics with more structured reading of figures, enabling consistent extraction of numeric values from plotted trajectories. In our setting, this includes both improved access to section-structured text and explicit mechanisms for mapping visual signals (e.g., plot markers) to calibrated values. A representative example of this figure-reading

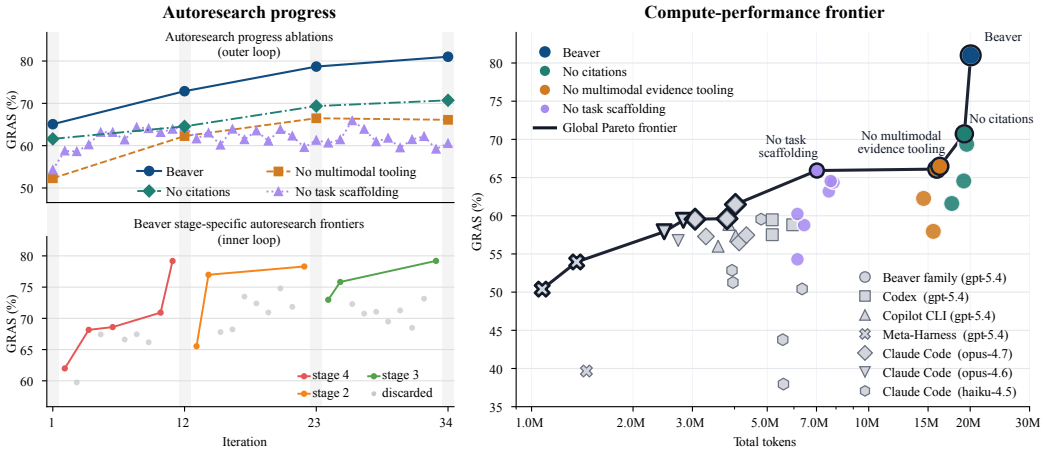


Figure 5: Left: **Autoresearch progress for Beaver and its three ablations.** The upper panel shows all-stage evaluations over autoresearch iterations, including Beaver and three ablations. The lower panel shows Beaver 's accepted stage-specific autoresearch frontiers (inner loop), with gray bands indicating the all-stage checkpoint slots (outer loop). Right: **Compute–performance frontier.** We plot GRAS against total token usage to compare how efficiently each harness converts agent compute into curation quality. Gray markers denote repeated runs of strong baselines across harnesses and backend models, while colored markers denote Beaver and its ablated variants. The connected line shows the empirical Pareto frontier, highlighting the best observed score at each token budget.

pipeline is provided in Sec. A.1. Second, a set of changes **preserve stagewise analysis families and carried-forward context**. These address failures where distinct clinical rows—such as pooled groups, subgroups, or comparison families—are prematurely merged or overwritten. The revised harness maintains these families explicitly across stages, ensuring that downstream extraction operates on the correct endpoint-level context. In practice, this means delaying aggregation and preserving identifiers long enough to recover the appropriate structured records. The core mechanism is formalized in Algorithm 2. Third, we observe improvements from **stronger deterministic postprocessing**. These changes enable reliable recovery of derived outcome attributes when sufficient local evidence is already present in the extracted row. Rather than requiring the model to emit fully normalized outputs, the harness performs lightweight normalization and arithmetic (e.g., computing `value` or `percentchange`) to match schema expectations. A concrete example is shown in Sec. A.1.

Failure Mode Analysis. Comparing Beaver against strong baseline agents and against ablations surfaces three contrasting case types. First, baseline agents miss **multi-visit figure trajectories**: they typically settle for one row per arm at the final visit, while Beaver recovers calibrated per-visit rows aligned to gold. On PMID 27756421, Beaver emits 80 rows matching the gold reference layout while the strongest baseline emits only 15 final-visit rows, producing a $+0.52$ per-paper GRAS gap (Case A in Sec. A.2). Second, removing individual Beaver components causes targeted regressions: stripping task scaffolding inflates extraction to the wrong endpoint family (PMID 22291741, -0.54), removing provenance grounding collapses subgroup row families (PMID 31884472, -0.40), and removing multimodal tooling reverts plot reading to imprecise visual estimates (PMID 27756421, -0.32); details in Case B of Sec. A.2. Third, Beaver still collapses **repeated analysis-population variants** (e.g., FAS, observed-cases, completers) into a single row per (subgroup, endpoint), as on PMID 29154277 where GRAS remains 0.48; we treat this direction as the priority growth area (Case C in Sec. A.2).

7 Related Work

Existing datasets relevant to our setting fall into several adjacent but incomplete categories. AxCell [14] and SciDaSynth [13] are among the closest examples of extracting structured results from scientific papers, but they target lighter-weight result extraction settings than the rich curation schema studied here. MatViX [6] is the closest conceptual multimodal extraction benchmark because it explicitly integrates evidence across text, tables, and figures, but it targets structured JSON extraction

in a different scientific domain. CTE [32], PubTables-1M [17], SciTSR [23], and TabLeX [33] are strong scientific-table benchmarks for contextualized extraction, table detection, structure recovery, or table-centric content extraction, but they do not evaluate end-to-end curation into a normalized record schema. Scientific information extraction benchmarks such as SciERC [34], SciREX [35], SciER [24], SciNLP [36], EBM-NLP [37], and BioRED [38] evaluate entity extraction, relation extraction, contribution extraction, measurement extraction, or biomedical evidence extraction from scientific text, but their targets are primarily entity-relation structures, keyphrases, contribution graphs, or span-level annotations rather than multimodal schema-normalized curation records. SciClaimEval [25] and cPAPERS [39] are also multimodal and paper-grounded, yet their targets are claim verification and question answering rather than structured table construction. Taken together, these benchmarks are highly relevant, but none directly evaluates full-paper multimodal scientific curation into a rich production-style schema with expert-curated reference data.

Related method work is closer in spirit to our harness design than to the benchmark itself. Karpathy's `autoresearch` project [29] provides the practical starting point for treating iterative research as a filesystem-backed agent loop, while Meta-Harness [19] is the closest direct methodological comparator because it also treats the harness as the object of optimization. Natural-Language Agent Harnesses [20] and Harness-Native Software Engineering [21] further support the view that the control logic and runtime environment around the model are first-class methodological choices, and recent engineering accounts likewise argue that harness design materially affects long-horizon agent behavior [22]. More broadly, recent work has explored reflective self-improvement [40, 41], text-based optimization [42, 43], evolutionary or zeroth-order search [44, 45], and prompt or pipeline optimization frameworks [46, 47]. As emphasized in Meta-Harness's related-work framing, many of these methods operate with shorter-horizon or more compressed feedback than full harness development. In contrast, Beaver focuses on multimodal scientific curation, combines staged extraction with multimodal evidence artifacts, and grounds revision in archived outputs, provenance traces, evaluation artifacts, diagnosis artifacts, and human-reviewed winning runs.

Discussion

Our results show that harness design matters in scientific curation from full papers with multimodal evidence. At the same time, the current study leaves several important open directions.

One is scaling the autoresearch loop further, both to test whether performance continues to improve beyond the current 34-iteration horizon and to study when returns begin to flatten. Another is more parallel development. In principle, stage-specific autoresearch can be run concurrently, with cross-stage transferred checkpoints used to test whether local gains compose cleanly at the system level. A third direction is broader generalization. The framework developed here is motivated by scientific curation from multimodal sources, but the underlying pattern of staged task structure, evidence-access tooling, and artifact-grounded harness revision should transfer to other multi-step agent tasks that require structured outputs rather than free-form answers.

More broadly, the main positive impact of this line of work is improved scientific curation efficiency. Many discovery workflows depend on extracting normalized evidence from large volumes of literature, and multimodal papers are a persistent bottleneck because the relevant evidence is distributed across text, tables, and figures. A stronger harness for this setting could reduce manual effort and accelerate downstream analysis. At the same time, this is also a setting where auditability matters: errors in normalized scientific records can propagate into later decisions. For that reason, we view human review, grounded diagnosis, and explicit evidence linkage not as optional extras, but as part of the responsible use of agentic curation systems.

References

[1] Fan Bai, Junmo Kang, Gabriel Stanovsky, Dayne Freitag, Mark Dredze, and Alan Ritter. Schema-driven information extraction from heterogeneous tables. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 10252–10273, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.600. URL https://aclanthology.org/2024.findings-emnlp.600/.

[2] Vishakh Padmakumar, Joseph Chee Chang, Kyle Lo, Doug Downey, and Aakanksha Naik. Intent-aware schema generation and refinement for literature review tables. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 23450–23472, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-335-7. doi: 10.18653/v1/2025.findings-emnlp.1274. URL https://aclanthology.org/2025.findings-emnlp.1274/.

[3] Phyllis Chan, Kirill Peskov, and Xuyang Song. Applications of model-based meta-analysis in drug development: Chan, peskov and song. *Pharmaceutical Research*, 39(8):1761–1777, 2022.

[4] Akari Asai, Jacqueline He, Rulin Shao, Weijia Shi, Amanpreet Singh, Joseph Chee Chang, Kyle Lo, Luca Soldaini, Sergey Feldman, Mike D'Arcy, et al. Synthesizing scientific literature with retrieval-augmented language models. *Nature*, pages 1–7, 2026.

[5] Satanu Ghosh, Neal Brodnik, Carolina Frey, Collin Holgate, Tresa Pollock, Samantha Daly, and Samuel Carton. Toward reliable ad-hoc scientific information extraction: A case study on two materials dataset. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Findings of the Association for Computational Linguistics: ACL 2024*, pages 15109–15123, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-acl.897. URL https://aclanthology.org/2024.findings-acl.897/.

[6] Ghazal Khalighinejad, Sharon Scott, Ollie Liu, Kelly L. Anderson, Rickard Stureborg, Aman Tyagi, and Bhuwan Dhingra. Matvix: Multimodal information extraction from visually rich articles, 2024. URL https://arxiv.org/abs/2410.20494.

[7] Zhiyong Lu and Lynette Hirschman. Biocuration workflows and text mining: overview of the biocreative 2012 workshop track ii. *Database*, 2012:bas043, 2012.

[8] Cecilia N Arighi, Phoebe M Roberts, Shashank Agarwal, Sanmitra Bhattacharya, Gianni Cesareni, Andrew Chatr-Aryamontri, Simon Clematide, Pascale Gaudet, Michelle Gwinn Giglio, Ian Harrow, et al. Biocreative iii interactive task: an overview. *BMC bioinformatics*, 12 (Suppl 8):S4, 2011.

[9] Thomas C. Wiegers, Allan Peter Davis, Jolene Wiegers, Daniela Sciaky, Fern Barkalow, Brent Wyatt, Melissa Strong, Roy McMorran, Sakib Abrar, and Carolyn J. Mattingly. Integrating AI-powered text mining from PubTator into the manual curation workflow at the comparative toxicogenomics database. *Database*, 2025:baaf013, 2025. doi: 10.1093/database/baaf013. URL https://doi.org/10.1093/database/baaf013.

[10] Certara, Inc. Certara reports fourth quarter 2025 financial results; provides full year 2026 guidance. SEC Exhibit 99.1 earnings release, February 2026. URL https://www.sec.gov/Archives/edgar/data/1827090/000182709026000008/q42025earningsreleaseex99.htm.

[11] Yoonjoo Lee, Kyungjae Lee, Sunghyun Park, Dasol Hwang, Jaehyeon Kim, Hong-in Lee, and Moontae Lee. Qasa: advanced question answering on scientific articles. In *International Conference on Machine Learning*, pages 19036–19052. PMLR, 2023.

[12] Pradeep Dasigi, Kyle Lo, Iz Beltagy, Arman Cohan, Noah A. Smith, and Matt Gardner. A dataset of information-seeking questions and answers anchored in research papers. In Kristina Toutanova, Anna Rumshisky, Luke Zettlemoyer, Dilek Hakkani-Tur, Iz Beltagy,

Steven Bethard, Ryan Cotterell, Tanmoy Chakraborty, and Yichao Zhou, editors, *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4599–4610, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.365. URL https://aclanthology.org/2021.naacl-main.365/.

[13] Xingbo Wang, Samantha L. Huey, Rui Sheng, Saurabh Mehta, and Fei Wang. Scidasynth: Interactive structured data extraction from scientific literature with large language model. *Campbell Systematic Reviews*, 21(4):e70073, 2025. doi: https://doi.org/10.1002/cl2.70073. URL https://onlinelibrary.wiley.com/doi/abs/10.1002/cl2.70073.

[14] Marcin Kardas, Piotr Czapla, Pontus Stenetorp, Sebastian Ruder, Sebastian Riedel, Ross Taylor, and Robert Stojnic. AxCell: Automatic extraction of results from machine learning papers. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8580–8594, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.692. URL https://aclanthology.org/2020.emnlp-main.692/.

[15] Zifeng Wang, Lang Cao, Qiao Jin, Joey Chan, Nicholas Wan, Behdad Afzali, Hyun-Jin Cho, Chang-In Choi, Mehdi Emamverdi, Manjot K Gill, et al. A foundation model for human-ai collaboration in medical literature mining. *Nature communications*, 16(1):8361, 2025.

[16] Yilun Zhao, Chengye Wang, Chuhan Li, and Arman Cohan. Can multimodal foundation models understand schematic diagrams? an empirical study on information-seeking QA over scientific papers. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar, editors, *Findings of the Association for Computational Linguistics: ACL 2025*, pages 18598–18631, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-256-5. doi: 10.18653/v1/2025.findings-acl.957. URL https://aclanthology.org/2025.findings-acl.957/.

[17] Brandon Smock, Rohith Pesala, and Robin Abraham. Pubtables-1m: Towards comprehensive table extraction from unstructured documents, 2021. URL https://arxiv.org/abs/2110.00061.

[18] David Wadden, Shanchuan Lin, Kyle Lo, Lucy Lu Wang, Madeleine van Zuylen, Arman Cohan, and Hannaneh Hajishirzi. Fact or fiction: Verifying scientific claims. In Bonnie Webber, Trevor Cohn, Yulan He, and Yang Liu, editors, *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7534–7550, Online, November 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.emnlp-main.609. URL https://aclanthology.org/2020.emnlp-main.609/.

[19] Meta-harness: End-to-end optimization of model harnesses, 2026. URL https://arxiv.org/abs/2603.28052.

[20] Natural-language agent harnesses, 2026. URL https://arxiv.org/abs/2603.25723.

[21] Chaitanya S. and collaborators. Harness-native software engineering: The control plane of coding agents, 2026. URL https://research.chaitanya.science/papers/harness-native-software-engineering.

[22] Anthropic. Harness design for long-running application development, mar 2026. URL https://www.anthropic.com/engineering/harness-design-long-running-apps.

[23] Zewen Chi, Heyan Huang, Heng-Da Xu, Houjin Yu, Wanxuan Yin, and Xian-Ling Mao. Complicated table structure recognition, 2019. URL https://arxiv.org/abs/1908.04729.

[24] Qi Zhang, Zhijia Chen, Huitong Pan, Cornelia Caragea, Longin Jan Latecki, and Eduard Dragut. SciER: An entity and relation extraction dataset for datasets, methods, and tasks in scientific documents. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen, editors, *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 13083–13100, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.726. URL https://aclanthology.org/2024.emnlp-main.726/.

[25] Xanh Ho, Yun-Ang Wu, Sunisth Kumar, Tian Cheng Xia, Florian Boudin, Andre Greiner-Petter, and Akiko Aizawa. Sciclaimeval: Cross-modal claim verification in scientific papers, 2026. URL https://arxiv.org/abs/2602.07621.

[26] Xu Zhong, Elaheh ShafieiBavani, and Antonio Jimeno Yepes. Image-based table recognition: Data, model, and evaluation. In *Computer Vision – ECCV 2020*, 2020. URL https://arxiv.org/abs/1911.10683.

[27] Brandon Smock, Rohith Pesala, and Robin Abraham. Grits: Grid table similarity metric for table structure recognition, 2022. URL https://arxiv.org/abs/2203.12555.

[28] Nick Ferguson, Josh Pennington, Narek Beghian, Aravind Mohan, Douwe Kiela, Sheshansh Agrawal, and Thien Hang Nguyen. Extractbench: A benchmark and evaluation methodology for complex structured extraction, 2026. URL https://arxiv.org/abs/2602.12247.

[29] Andrej Karpathy. autoresearch. GitHub repository, 2026. URL https://github.com/karpathy/autoresearch.

[30] National Library of Medicine. Pmc open access subset. https://pmc.ncbi.nlm.nih.gov/tools/openftlist/, 2003. Last modified September 11, 2025.

[31] Mike A. Merrill, Alexander G. Shaw, Nicholas Carlini, Boxuan Li, Harsh Raj, Ivan Bercovich, Lin Shi, Jeong Yeon Shin, Thomas Walshe, E. Kelly Buchanan, Junhong Shen, Guanghao Ye, Haowei Lin, Jason Poulos, Maoyu Wang, Marianna Nezhurina, Jenia Jitsev, Di Lu, Orfeas Menis Mastromichalakis, Zhiwei Xu, Zizhao Chen, Yue Liu, Robert Zhang, Leon Liangyu Chen, Anurag Kashyap, Jan-Lucas Uslu, Jeffrey Li, Jianbo Wu, Minghao Yan, Song Bian, Vedang Sharma, Ke Sun, Steven Dillmann, Akshay Anand, Andrew Lanpouthakoun, Bardia Koopah, Changran Hu, Etash Guha, Gabriel H. S. Dreiman, Jiacheng Zhu, Karl Krauth, Li Zhong, Niklas Muennighoff, Robert Amanfu, Shangyin Tan, Shreyas Pimpalgaonkar, Tushar Aggarwal, Xiangning Lin, Xin Lan, Xuandong Zhao, Yiqing Liang, Yuanli Wang, Zilong Wang, Changzhi Zhou, David Heineman, Hange Liu, Harsh Trivedi, John Yang, Junhong Lin, Manish Shetty, Michael Yang, Nabil Omi, Negin Raoof, Shanda Li, Terry Yue Zhuo, Wuwei Lin, Yiwei Dai, Yuxin Wang, Wenhao Chai, Shang Zhou, Dariush Wahdany, Ziyu She, Jiaming Hu, Zhikang Dong, Yuxuan Zhu, Sasha Cui, Ahson Saiyed, Arinbjörn Kolbeinsson, Jesse Hu, Christopher Michael Rytting, Ryan Marten, Yixin Wang, Alex Dimakis, Andy Konwinski, and Ludwig Schmidt. Terminal-bench: Benchmarking agents on hard, realistic tasks in command line interfaces, 2026. URL https://arxiv.org/abs/2601.11868.

[32] A Gemelli, E Vivoli, S Marinai, et al. Cte: A dataset for contextualized table extraction. In *CEUR WORKSHOP PROCEEDINGS*, volume 3365, pages 197–208. CEUR-WS, 2023.

[33] Harsh Desai, Pratik Kayal, and Mayank Singh. Tablex: A benchmark dataset for structure and content information extraction from scientific tables. *CoRR*, abs/2105.06400, 2021. URL https://arxiv.org/abs/2105.06400.

[34] Yi Luan, Luheng He, Mari Ostendorf, and Hannaneh Hajishirzi. Multi-task identification of entities, relations, and coreference for scientific knowledge graph construction. In Ellen Riloff, David Chiang, Julia Hockenmaier, and Jun'ichi Tsujii, editors, *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3219–3232, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1360. URL https://aclanthology.org/D18-1360/.

[35] Sarthak Jain, Madeleine van Zuylen, Hannaneh Hajishirzi, and Iz Beltagy. SciREX: A challenge dataset for document-level information extraction. In Dan Jurafsky, Joyce Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7506–7516, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.670. URL https://aclanthology.org/2020.acl-main.670/.

[36] Decheng Duan, Jitong Peng, Yingyi Zhang, and Chengzhi Zhang. SciNLP: A domain-specific benchmark for full-text scientific entity and relation extraction in NLP. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 14473–14486, Suzhou, China, November 2025. Association for Computational

Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.732. URL https://aclanthology.org/2025.emnlp-main.732/.

[37] Benjamin Nye, Junyi Jessy Li, Roma Patel, Yinfei Yang, Iain Marshall, Ani Nenkova, and Byron Wallace. A corpus with multi-level annotations of patients, interventions and outcomes to support language processing for medical literature. In Iryna Gurevych and Yusuke Miyao, editors, *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 197–207, Melbourne, Australia, July 2018. Association for Computational Linguistics. doi: 10.18653/v1/P18-1019. URL https://aclanthology.org/P18-1019/.

[38] Ling Luo, Po-Ting Lai, Chih-Hsuan Wei, Cecilia N Arighi, and Zhiyong Lu. Biored: a rich biomedical relation extraction dataset. *Briefings in Bioinformatics*, 23(5):bbac282, 2022.

[39] Anirudh Sundar, Jin Xu, William Gay, Christopher Richardson, and Larry Heck. cpapers: A dataset of situated and multimodal interactive conversations in scientific papers. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024) Datasets and Benchmarks Track*, 2024. doi: 10.52202/079017-2119. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/7a19a9d527ed544d1272f07b0f8f934e-Abstract-Datasets_and_Benchmarks_Track.html.

[40] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Shashank Gupta, Bodhisattwa Prasad Majumder, Katherine Hermann, Sean Welleck, Amir Yazdanbakhsh, and Peter Clark. Self-refine: Iterative refinement with self-feedback, 2023. URL https://arxiv.org/abs/2303.17651.

[41] Qin Liu, Wenxuan Zhou, Nan Xu, James Y Huang, Fei Wang, Sheng Zhang, Hoifung Poon, and Muhao Chen. Metascale: Test-time scaling with evolving meta-thoughts. *arXiv preprint arXiv:2503.13447*, 2025.

[42] Mert Yuksekgonul, Federico Bianchi, Joseph Boen, Sheng Liu, Zhi Huang, Carlos Guestrin, and James Zou. Textgrad: Automatic "differentiation" via text, 2024. URL https://arxiv.org/abs/2406.07496.

[43] Yoonho Lee, Joseph Boen, and Chelsea Finn. Feedback descent: Open-ended text optimization via pairwise comparison, 2025. URL https://arxiv.org/abs/2511.07919.

[44] Mert Cemri, Shubham Agrawal, Akshat Gupta, Shu Liu, Audrey Cheng, Qiuyang Mang, Ashwin Naren, Lutfi Eren Erdogan, Koushik Sen, Matei Zaharia, Alex Dimakis, and Ion Stoica. Adaevolve: Adaptive llm driven zeroth-order optimization, 2026. URL https://arxiv.org/abs/2602.20133.

[45] Alexander Novikov, Ngân Vũ, Marvin Eisenberger, Emilien Dupont, Po-Sen Huang, Adam Zsolt Wagner, Sergey Shirobokov, Borislav Kozlovskii, Francisco J. R. Ruiz, Abbas Mehrabian, M. Pawan Kumar, Abigail See, Swarat Chaudhuri, George Holland, Alex Davies, Sebastian Nowozin, Pushmeet Kohli, and Matej Balog. Alphaevolve: A coding agent for scientific and algorithmic discovery, 2025. URL https://arxiv.org/abs/2506.13131.

[46] Lakshya A Agrawal, Shangyin Tan, Dilara Soylu, Noah Ziems, Rishi Khare, Krista Opsahl-Ong, Arnav Singhvi, Herumb Shandilya, Michael J Ryan, Meng Jiang, Christopher Potts, Koushik Sen, Alexandros G. Dimakis, Ion Stoica, Dan Klein, Matei Zaharia, and Omar Khattab. Gepa: Reflective prompt evolution can outperform reinforcement learning, 2026. URL https://arxiv.org/abs/2507.19457.

[47] Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T. Joshi, Hanna Moazam, Heather Miller, Matei Zaharia, and Christopher Potts. Dspy: Compiling declarative language model calls into self-improving pipelines, 2023. URL https://arxiv.org/abs/2310.03714.

A Appendix

Ablation Autoresearch Trajectories. Fig. 6 shows the optimization trajectories for each ablated harness. For the no-provenance-traces and no-multimodal-tooling variants, the top panels show outer-loop improvements over the autoresearch budget, while the bottom panels show stage-specific inner-loop candidates, distinguishing retained frontiers from discarded attempts. The no-task-scaffolding variant lacks stage-localized frontiers because the harness no longer decomposes execution into stable stages. These trajectories illustrate that autoresearch can still find improvements under ablations, but the search becomes less effective or less structured when key harness components are removed.

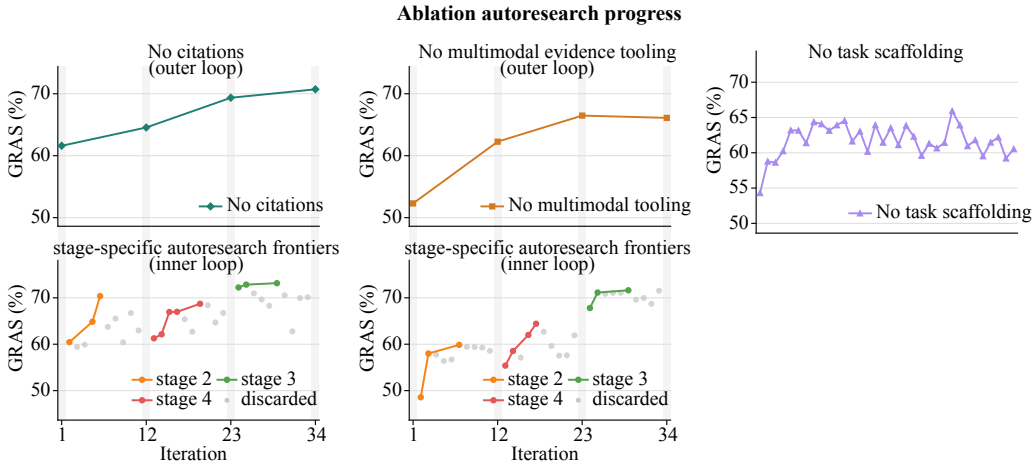


Figure 6: Autoresearch trajectories for ablated harnesses. Top panels show outer-loop progress over the autoresearch budget, while bottom panels show stage-specific inner-loop candidates, with retained frontier updates separated from discarded attempts. Without task scaffolding, the method lacks stable stage-localized workspaces, so only the aggregate progress trajectory is shown.

A.1 Representative Harness Changes

This subsection provides three representative harness changes discovered by Beaver. Taken together, these changes clarify why autoresearch helps in this setting: the gains come from improving how the harness preserves scientific structure, accesses multimodal evidence, and normalizes extracted values.

Figure-grounded evidence access. A representative change to the figure-reading harness replaced general visual inspection with explicit axis calibration and narrow-band marker detection. The resulting workflow first anchors the plot axes in pixel space, converts between pixel rows and numeric values, and only then reads candidate points near the relevant timepoint. The mechanism is illustrated by the following representative snippet:

```
# gray: single-channel image intensity array used for scanning
# x_ticks: x-pixel centers for labeled visits or timepoints
# THRESHOLD: darkness cutoff for axis, tick, and marker evidence
# AXIS_* and TICK_*: approximate search windows around the plot axes
# *_DARK_MIN: minimum dark-pixel evidence for accepting a local candidate

AXIS_COL_MIN = plot_left_col
AXIS_COL_MAX = plot_left_col + axis_search_width
AXIS_SCAN_ROW_MIN, AXIS_SCAN_ROW_MAX = plot_top_row, plot_bottom_row
TICK_ROW_MIN, TICK_ROW_MAX = plot_top_row, plot_bottom_row
TICK_BAND_WIDTH = axis_tick_band_width
MARKER_BAND_RADIUS = marker_scan_radius
AXIS_DARK_MIN = axis_dark_min
TICK_DARK_MIN = tick_dark_min
```

```
MARKER_DARK_MIN = marker_dark_min

# Search candidate columns for the y-axis line.
axis_col = None
for c in range(AXIS_COL_MIN, AXIS_COL_MAX):
    dark = np.sum(gray[AXIS_SCAN_ROW_MIN:AXIS_SCAN_ROW_MAX, c] < THRESHOLD)
    if dark > AXIS_DARK_MIN:
        axis_col = c

# Search just left of the detected axis for y-tick candidates.
tick_rows = []
for r in range(TICK_ROW_MIN, TICK_ROW_MAX):
    dark = np.sum(gray[r, axis_col-TICK_BAND_WIDTH:axis_col] < THRESHOLD)
    if dark >= TICK_DARK_MIN:
        tick_rows.append(r)

pixels_per_unit = (px_bottom_tick - px_top_tick) \
    / (val_top_tick - val_bottom_tick)

def ypx_to_val(y_px):
    return (y_zero_px - y_px) / pixels_per_unit

# Scan a narrow band around each x-position for candidate data markers.
for label, xcol in x_ticks.items():
    x_lo = max(0, xcol - MARKER_BAND_RADIUS)
    x_hi = min(w, xcol + MARKER_BAND_RADIUS + 1)
    candidate_rows = []
    for r in range(plot_top_row, plot_bottom_row):
        dark_count = np.sum(gray[r, x_lo:x_hi] < THRESHOLD)
        if dark_count >= MARKER_DARK_MIN:
            candidate_rows.append((r, dark_count))

    groups = group_consecutive(candidate_rows, max_gap=row_group_gap)
    for g in groups:
        center_row = best_row_in_group(g)
        y_val = ypx_to_val(center_row)
        print(f"{label}: row {center_row}, y = {y_val:.2f}")
```

This change matters because figure-driven rows are often lost even when the correct figure has been found: the real bottleneck is turning a visually localized trace into a stable numeric readout aligned to the correct visit and arm.

Family preservation. A second class of changes are applied to how `baseline` related attributes are emitted before endpoint-level expansion. This family-preservation logic reduces a recurring class of failures where pooled rows, subgroup rows, or comparison families are merged too early and cannot be recovered downstream. Due to length considerations, we present the pseudocode below instead of a full code listing:

Deterministic postprocessing. A third class of changes adds deterministic same-row cleanup after the raw extraction. Rather than relying on the model to emit every derived value directly, the harness now fills outcome attributes such as `value`, `percentchange`, and `n.observed` when the same row already contains enough local evidence to do so consistently. The following excerpt from the postprocessor shows the core mechanism:

```
if baseline is not None and change is not None:
    if _is_blank(row.get("value")):
        row["value"] = _format_decimal(baseline + change)
    if baseline != 0 and _is_blank(row.get("percentchange")):
        row["percentchange"] = _format_decimal(
```

Algorithm 2 Stage-n Row-Family Preservation

Require: paper p, carried-forward row $\hat{r}^{(n-1)} \in \hat{Y}^{(n-1)}$, evidence artifacts $(\mathcal{T}(p), \mathcal{B}(p), \mathcal{G}(p), \mathcal{A}(p))$
Ensure: emitted row set $\mathcal{R}^{(n)}(\hat{r}^{(n-1)}) \subseteq \hat{Y}^{(n)}$ over $\mathcal{F}^{(n)}$

$\mathcal{R}^{(n)}(\hat{r}^{(n-1)}) \leftarrow \emptyset$
$\mathcal{B}_{\text{base}}(p, \hat{r}^{(n-1)}) \leftarrow$ baseline-characteristics table candidates for the endpoint in $\hat{r}^{(n-1)}$
$\mathcal{B}_{\text{comp}}(p, \hat{r}^{(n-1)}) \leftarrow$ later comparison-table candidates for the endpoint in $\hat{r}^{(n-1)}$
$\mathcal{B}^{\star}(p, \hat{r}^{(n-1)}) \leftarrow \mathcal{B}_{\text{base}}(p, \hat{r}^{(n-1)})$ if both candidate sets are non-empty; otherwise use the available candidate set
$\Pi(\hat{r}^{(n-1)}, p) \leftarrow$ explicit analysis families supported by the paper for $\hat{r}^{(n-1)}$
if $|\Pi(\hat{r}^{(n-1)}, p)| = 0$ **then**
    $\Pi(\hat{r}^{(n-1)}, p) \leftarrow \{f_0\}$ ▷ default carried-forward family
end if
for all $f \in \Pi(\hat{r}^{(n-1)}, p)$ **do**
    initialize $\hat{r}_f^{(n)}$ from the carried-forward attributes in $\hat{r}^{(n-1)}$
    populate `baseline`$(\hat{r}_f^{(n)})$ and `baseline.sd`$(\hat{r}_f^{(n)})$ from $\mathcal{B}^{\star}(p, \hat{r}^{(n-1)})$ for family f
    if `n.randomized`$(\hat{r}^{(n-1)})$ is blank $\wedge$ no exact-family denominator is reported in $(\mathcal{T}(p), \mathcal{B}(p), \mathcal{G}(p), \mathcal{A}(p))$ **then**
        preserve blank `n.randomized`$(\hat{r}_f^{(n)})$
    else
        set `n.randomized`$(\hat{r}_f^{(n)})$ from the exact-family denominator when available
    end if
    carry the canonical endpoint spelling and family identity from $\hat{r}^{(n-1)}$ into $\hat{r}_f^{(n)}$
    $\mathcal{R}^{(n)}(\hat{r}^{(n-1)}) \leftarrow \mathcal{R}^{(n)}(\hat{r}^{(n-1)}) \cup \{\hat{r}_f^{(n)}\}$
end for
return $\mathcal{R}^{(n)}(\hat{r}^{(n-1)})$

```
            (change / baseline) * Decimal("100")
        )

if n_randomized is not None and _is_blank(row.get("n.observed")):
    row["n.observed"] = _format_decimal(n_randomized)
```

A.2 Failure mode case studies

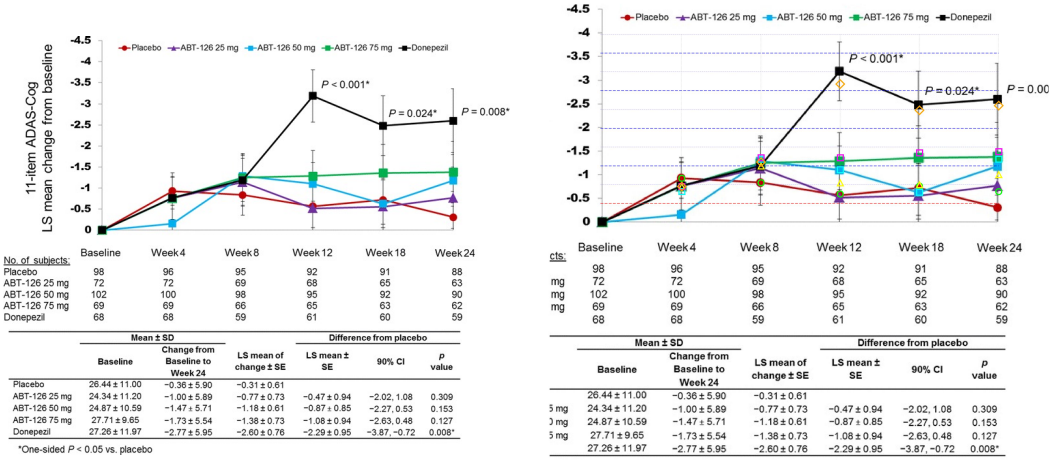
The case studies below elaborate the three contrasting case types summarized in Sec. 6, with per-paper GRAS scores averaged across the three replicate runs of each family. Table 4 collects the headline numbers for the four representative papers discussed in this section.

Table 4: Per-paper GRAS for the four papers used as failure-mode case studies, averaged across three replicate runs per family. The "Baseline Agent (best)" column reports the strongest baseline agent; "Baseline Agents (avg)" averages all baseline agents.

PMID	Beaver	Baseline Agent (best)	Baseline Agents (avg)	No task scaffolding	No provenances	No multimodal evidence tooling
27756421	0.821	0.302 (Codex)	0.271	0.310	0.580	0.503
22291741	1.000	0.875 (Copilot)	0.716	0.460	0.714	0.714
31884472	0.755	0.720 (Codex)	0.636	0.781	0.356	0.364
29154277	0.478	0.317 (Codex)	0.282	0.403	0.543	0.484

Case A: Baselines fail, Beaver succeeds (PMID 27756421). This trial reports its primary ADAS-Cog outcomes as a multi-visit figure (Fig. 7 left), with per-visit means as plotted markers and per-visit subject counts in a row beneath the plot. The gold reference contains 80 rows spanning all visits,

treatment arms, and endpoint variants. Every baseline agent reports only the final-visit row per arm: codex emits 15 rows, meta-harness 10, and copilot variants between 10 and 30, all clustered around GRAS ≈ 0.27. Beaver emits 80 rows aligned to the gold reference and reaches 0.821, a $+0.52$ gap over the strongest baseline. The supporting provenances explicitly reference axis-calibrated gridline overlays produced by the multimodal evidence tooling (e.g., a per-row note "LS mean change estimated from calibrated gridline reading of the plotted marker"); Fig. 7 shows the original figure beside the version produced by Beaver using multimodal evidence tooling.



(a) Original Fig. 2 in the paper.

(b) The same figure after Beaver's multimodal evidence tooling.

Figure 7: Case A (PMID 27756421). Panel (a) is the original ADAS-Cog change-from-baseline trajectory plotted across visits per treatment arm; panel (b) is the artifact written by Beaver's multimodal evidence tooling, with calibrated y-axis gridlines overlaid on the marker band and the per-visit (mean, SE, n) read-out re-emitted as a structured table beneath the plot. Off-the-shelf baselines and the no-multimodal ablation either skip the figure or estimate values without calibration; Beaver uses the gridline overlay to produce the 80 per-visit rows that align with gold.

Case B: Ablations fail, Beaver succeeds. We pick one PMID per ablation family that exposes the load-bearing role of the corresponding Beaver component.

No-task-scaffolding (PMID 22291741). The trial's primary endpoint is MMSE at week 78 across three arms (placebo and two tramiprosate doses), as fixed by the CONSORT diagram in Fig. 8; the gold reference is just three rows. Beaver emits exactly those three rows and matches gold on every scored field (1.000). Removing task scaffolding causes the agent to drop the staged endpoint-family carry-forward and instead enumerate every visit $\times$ endpoint pair from the body's ADAS-Cog tables (72 rows), producing a 0.460 score.

No-provenances (PMID 31884472). The gold layout has 11 rows spread across four dose levels: per dose, one primary "ADAS-cog-11 | MMSE" row plus one or two "ADAS-cog-11"-only subgroup rows distinguished by which subset of randomized subjects had each rating. Beaver preserves this row family (12 rows; 0.755) by anchoring each subgroup row in a distinct provenance evidence span (`s2c001` through `s2c021`). Stripping provenances collapses the output to four primary rows (0.356). The same paper is the strongest no-multimodal regression as well (0.364) because the subgroup decomposition is communicated in a small inset table whose structural reading benefits from multimodal tooling.

No-multimodal (PMID 27756421, stage 4). Re-using the trial from Case A, removing multimodal tooling does not eliminate figure reading entirely (the ablation still emits 50 rows), but it replaces the calibrated gridline reads shown in Fig. 7(b) with prose "visually estimated from the plotted marker" and loses the per-visit subject counts. GRAS drops from 0.821 to 0.503, confirming that the gain from multimodal tooling is predominantly numerical fidelity rather than figure detection.

Case C: Beaver still fails (PMID 29154277). The trial reports baseline cognitive scores under multiple analysis populations within a single subgroup: the gold reference contains 31 rows across

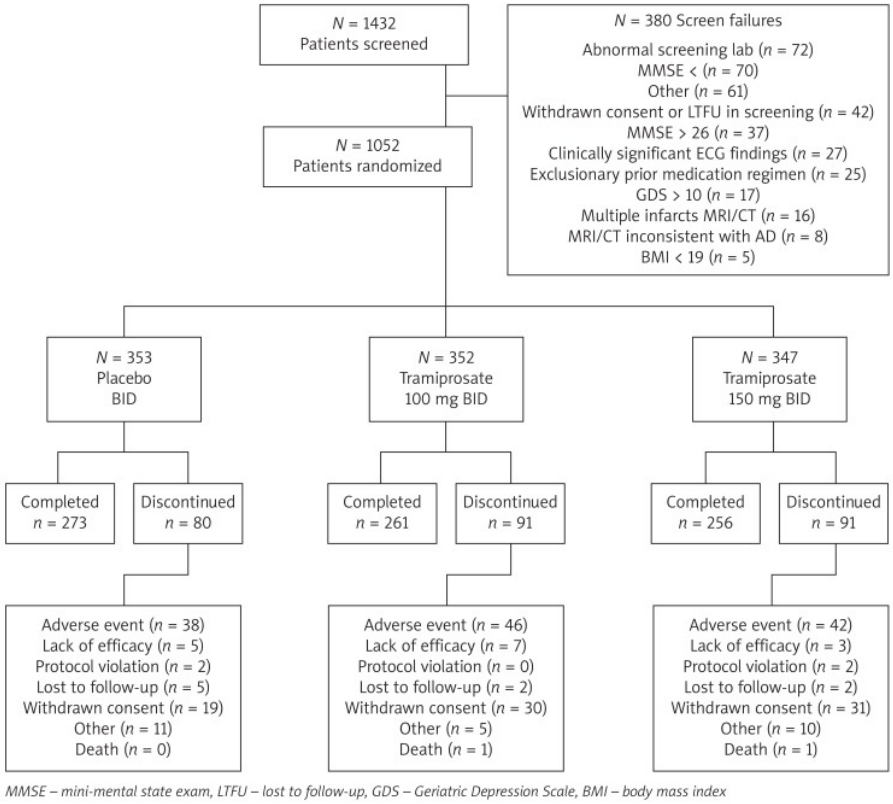


Figure 8: Case B / no-task-scaffolding (PMID 22291741). The paper's CONSORT diagram fixes the trial structure: 1052 patients randomized to placebo and two tramiprosate doses ($n = 353, 352, 347$), with MMSE at week 78 as the gold-relevant primary endpoint. Beaver emits the three matching rows; the no-scaffolding ablation instead enumerates 72 visit $\times$ endpoint rows from the ADAS-Cog tables because no prior-stage carry-forward keeps it on the MMSE-week-78 family.

two doses, two endpoints (ADAS-cog-11 and MMSE), and four subgroups, with each (subgroup, endpoint) cell reporting one to three baseline values from different analysis populations (e.g., FAS, observed-cases, completers). Beaver emits only 14 rows (one per (dose, subgroup, endpoint)), collapsing the analysis-population variants. GRAS is consistently 0.48 ± 0.02 across replicates. Closing this gap requires the harness to recognise the analysis-population axis as a fourth row-family dimension, which the current pipeline does not.

Taken together, these cases show that Beaver's headline gains over baselines come predominantly from figure trajectory reconstruction (Case A) and stagewise row-family preservation (Case B), and that the largest remaining gap is along an axis (analysis-population) that the current harness does not yet treat as a row-distinguishing field (Case C).

A.3 Evaluation Papers

Table 5 reports the 23 PMIDs in the evaluation set and marks the 8 papers used in the development subset inline.

A.4 Schema Details

Table 6 gives the 20 scored attributes used in the main experiments.

PMIDs	29154277 (✓), 29067345 (✓), 26064192 (✓), 31329216, 32508323, 27176461, 28550255, 27756421 (✓), 22291741 (✓), 29067304 (✓), 31884472 (✓), 30231896, 30134967, 22606044, 32568196, 31944580, 30309389, 29221491, 29695589, 18213383, 25755685 (✓), 22567095, 24423155

Table 5: PMIDs for the 23-paper evaluation set. A checkmark in parentheses marks membership in the 8-paper development subset.

Category	Attribute	Type	Description
Source	`source.number`	numerical-discrete	Unique ID number of the reference. Use the PMID if available; for other references, assign a unique number.
Study	`registry.study.id`	text-descriptive	Registry ID of the study, including the registry acronym; include multiple IDs when applicable.
Study	`indication`	text-categorical	Indication of the study.
Study	`treatment.duration`	numerical-continuous	Duration of treatment in weeks, corresponding to the end of regular dosing.
Study	`study.duration`	numerical-continuous	Total study duration in weeks, including follow-up.
Endpoint	`endpoint`	text-categorical	Name of the endpoint.
Arm	`n.randomized`	numerical-discrete	Number of patients randomized to the arm or stratum.
Treatment	`drug1`	text-categorical	Name of the first drug; for combinations, the experimental drug is listed first.
Treatment	`drug1.dose`	numerical-continuous	Dose of the first drug, standardized to a common unit across studies.
Treatment	`drug1.dose.unit`	text-descriptive	Standardized dose unit used for the first drug.
Endpoint	`endpoint.unit`	text-categorical	Unit used for baseline, value, and change after endpoint normalization.
Endpoint	`baseline`	numerical-continuous	Baseline value.
Endpoint	`baseline.sd`	numerical-continuous	Standard deviation for the baseline value.
Endpoint	`n.observed`	numerical-discrete	Number of patients analyzed for the endpoint value.
Endpoint	`value`	numerical-continuous	Mean or median endpoint value at the reported timepoint, normalized to the preferred unit.
Endpoint	`value.sd`	numerical-continuous	Standard deviation for the endpoint value.
Endpoint	`change`	numerical-continuous	Mean or median change from baseline at the reported timepoint.
Endpoint	`change.sd`	numerical-continuous	Standard deviation for change from baseline.
Endpoint	`percentchange`	numerical-continuous	Mean or median percent change from baseline at the reported timepoint.
Endpoint	`percentchange.sd`	numerical-continuous	Standard deviation for percent change from baseline.

Table 6: The 20 scored attributes used in the main experiments.