U2: CAN MODELS LEARN TO ASK THE UNSEEN AND THE UNSOLVABLE?

Anonymous authors
Paper under double-blind review

ABSTRACT

As models become increasingly capable, recent benchmarks have introduced challenging questions that require models to perform complex reasoning and utilize external tools, such as web search or coding sandboxes. However, these benchmarks are costly to construct and typically demand significant effort from human experts for question design and annotation. Consequently, these datasets are difficult to scale up and update over time, posing a challenge as models risk training on similar, static questions. To address this problem, we propose training a question generator that takes academic materials, such as research papers, as input and outputs complex questions requiring a comprehensive understanding of the source text. Specifically, we employ reinforcement learning (RL) and design several rewards to incentivize the model to generate increasingly difficult questions. The most critical of these is the "difficulty reward," which utilizes another powerful model to answer the generated questions during rollout. This solver model yields an answer distribution, where higher uncertainty indicates a more challenging question. Following training, we employ the generator to construct the Unseen and Unsolvable Benchmark (`U2-Bench`), composed of three 500-question tracks: (1) open-ended questions requiring comprehensive, citation-grounded reports; (2) fact-heavy non-STEM QA requiring extensive searching and reasoning; and (3) calculation-heavy STEM QA requiring source-grounded derivation and computation. Across these fixed evaluation sets, frontier models equipped with external tools retain substantial room for improvement. We further introduce `U2-Train`, a fully self-evolving setting in which the question generator, difficulty model, and target solver are all derived from the same base model. The generator produces questions that the base model itself finds difficult, and the validated questions are then used to train a fresh solver initialized from the same base model. The resulting GPT-OSS-20B model improves over the base by 4.8, 3.3, and 3.6 points on GAIA, FRAMES, and HLE.

1 INTRODUCTION

Recent advances in Large Language Models (LLMs) have rapidly transformed artificial intelligence (Wang et al., 2024; Hsiao et al., 2025; Shi et al., 2025; Qu et al., 2025). Frontier models such as GPT-5.5 (OpenAI, 2026), Gemini-3.1-Pro (Google DeepMind, 2026), and Claude-Sonnet-4.5 (Anthropic, 2025) increasingly saturate standard reasoning and knowledge benchmarks (Jimenez et al., 2023; Jain et al., 2024; Dekoninck et al., 2026). Evaluation has therefore shifted toward long-horizon, tool-augmented benchmarks such as GAIA (Mialon et al., 2023) and DeepResearch (Du et al., 2025), alongside expert-level closed-ended benchmarks such as HLE (Phan et al., 2025).

Despite their utility, these benchmarks face two critical bottlenecks: scalability and data contamination (Xu et al., 2025b; Sun et al., 2025). Expert-level, multi-hop datasets demand extensive manual curation, annotation, and verification (Xu et al., 2025a; Wang et al., 2026a), leaving them small, static, costly to refresh, and increasingly exposed to training-data leakage (Chen et al., 2025). We therefore need an automated, scalable mechanism for dynamically generating high-quality, verifiable, and exceptionally difficult evaluation tasks.

To address this challenge, we ask: *Can we train models to autonomously generate unseen questions that remain challenging as systems improve?* We treat difficulty as an operational moving target:

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

Table 1: Comparison of U2-Bench with recent expert-level and long-horizon benchmarks. Dynamic generation reduces marginal expert labor after substantial one-time training and verification costs.

Benchmark	Domain	Format	Scale	Dynamic	Construction Cost
GAIA (Mialon et al., 2023)	General / Tools	QA	∼450	✗	High (Human)
HLE (Phan et al., 2025)	Multi-domain	QA	3K	✗	**Very High** (Experts)
DeepResearch (Du et al., 2025)	Scientific / Web	Report	Hundreds	✗	**Very High** (Experts)
LiveResearch (Wang et al., 2026a)	Scientific / Web	Report	Hundreds	✓	High (Continuous)
BrowseComp (Wei et al., 2025)	Web	QA	1.3K	✗	High (Human)
DeepSearchQA (Gupta et al., 2026)	Multi-domain	QA	900	✗	High (Human)
FrontierScience (Wang et al., 2026b)	Scientific	Hybrid	∼700	✗	**Very High** (Experts)
U2-Bench	**Multi-domain**	**Hybrid**	**1,500**	✓	**Low marginal expert labor**

each benchmark version should expose substantial headroom for current systems and be refreshable as they improve, rather than contain questions that are impossible in principle. We train a generator that ingests raw academic literature—including recent, unindexed LaTeX preprints—and produces questions requiring retrieval, synthesis, or derivation from source-specific evidence. Rather than relying on supervised fine-tuning, we optimize the generator with Group Relative Policy Optimization (GRPO) (Shao et al., 2024).

Our RL framework combines formatting, lexical self-containment, factual grounding, and difficulty rewards. Self-containment suppresses prompt-local meta-references such as "according to this paper"; it does not require that every source-specific fact be recoverable from the question alone. During rollout, a strong solver receives the question but not an injected copy of the intended source paper, while retaining web-search and code tools. Retrieving the relevant paper and using its evidence are therefore intended capabilities rather than leakage. We measure the entropy of the solver's sampled answers to prioritize questions that remain difficult under this access policy. Entropy alone, however, cannot distinguish deep reasoning from omitted qualifiers, so we pair difficulty selection with a separate source-blind validity check. A verifier trained by supervised fine-tuning (SFT) on multi-agent labels independently gates factual consistency with the generation context.

Leveraging this trained generator, we construct the Unseen and Unsolvable Benchmark (U2-Bench), a dynamic, multi-domain evaluation suite comprising three distinct tracks: open-ended questions requiring comprehensive, citation-grounded research reports; non-STEM QA requiring extensive multi-document retrieval and logical deduction; and calculation-heavy STEM questions requiring source-grounded derivation and computation. Our comprehensive evaluations demonstrate that U2-Bench presents a formidable challenge to current state-of-the-art systems. Specifically, GPT-5.5 (OpenAI, 2026) achieves only 55.6% accuracy on non-STEM QA without tools, improving to 75.8% with coding and search; on calculation-heavy STEM QA, GPT-5.5 and GPT-5-mini achieve 77.4% and 46.4%, respectively. Smaller open-weight models such as Qwen3.6-35B-A3B (Qwen Team, 2026) and Mistral-Small-3.2-24B (Mistral AI, 2025) struggle to surpass 30.0% on non-STEM QA. As shown in Table 1, compared with existing expert-level and long-horizon benchmarks that suffer from static formats and prohibitively high human construction costs, U2-Bench uniquely provides a dynamic, low-cost, and scalable alternative.

Beyond evaluation, we introduce U2-Train, a fully self-evolving setting built from the same question-generation framework. Unlike U2-Bench, which uses a stronger external solver as its difficulty model, U2-Train derives the question generator, difficulty model, and target solver from the same base model. One instance is trained to generate complex questions, a separate instance identifies questions that expose the base model's own capability gaps, and the validated questions are used to train a fresh solver initialized from that base model. This design tests whether a model can improve by learning to ask questions that it cannot yet answer reliably. The resulting GPT-OSS-20B (Agarwal et al., 2025) solver improves over the base from 59.3 to 64.1 on GAIA (Mialon et al., 2023), 76.2 to 79.5 on FRAMES (Krishna et al., 2025), and 16.6 to 20.2 on HLE (Phan et al., 2025), corresponding to gains of 4.8, 3.3, and 3.6 points, respectively.

2 QUESTION GENERATOR

To automatically synthesize highly complex, verifiable, and previously unasked scientific questions, in this section, we propose training a model that takes academic papers as input and generates novel questions grounded in these sources.

2.1 QUESTION GENERATION VIA REINFORCEMENT LEARNING

Let $\mathcal{D}$ denote a corpus of academic literature formatted in LaTeX. For a given seed paper, we construct an input context $C = \mathcal{S} \cup \mathcal{R}$, where $\mathcal{S} = \{s_1, s_2, \ldots, s_k\}$ represents the raw LaTeX source code of k selected sections from the seed paper, and $\mathcal{R}$ denotes the corresponding LaTeX source code of contextual information extracted from related literature. Our objective is to learn a generator policy π_θ, parameterized by a LLM, that maps the context C to an output tuple y:

$$y \sim \pi_\theta(\cdot|C), \quad \text{where } y = (r, q), \tag{1}$$

and $r = [r_1, r_2, \ldots, r_T]$ represents the step-by-step thinking process, and q is the final synthesized question. The model focuses solely on question synthesis rather than answer generation. The generated questions q are bipartite in nature: (1) complex inquiries necessitating a long-form report, and (2) deterministic formats, including Multiple Choice Questions (MCQs) and short QA, which are suitable for exact-match evaluation.

To optimize the generator policy π_θ, we adopt Group Relative Policy Optimization (GRPO) (Shao et al., 2024). For each context C, the behavior policy samples N trajectories. GRPO maximizes a token-averaged clipped likelihood-ratio objective using group-normalized rewards and a KL penalty to the reference policy. Appendix A.2 gives the complete objective and defines the clipping radius, KL coefficient, and numerical stabilization.

2.2 REWARDS DESIGN

For a trajectory $y = (r, q)$, the paragraph-level factual gate multiplies a fixed weighted sum of the reward components active for its question format. Open-ended generation uses format, lexical self-containment, paper selection, and coverage; deterministic short-form generation retains format, self-containment, and section selection, replaces coverage with difficulty, and adds quantitative structure for calculation-heavy questions. Table 2 summarizes the components, with complete definitions in Appendix A.1.

Table 2: Reward components used by the question generator.

Reward	**Role**
Paragraph verification	Universal gate: any source-inconsistent reasoning paragraph sets $R(y) = 0$.
Format and selection	Enforce parseable tags and valid paper or section choices, including the seed paper.
Lexical self-containment	Penalize prompt-local references such as "in this paper" without requiring source-specific facts to be answerable without retrieval.
Coverage	For open-ended reports, reward technical depth, multi-paper breadth, and specificity.
Difficulty	For fact-heavy and calculation-heavy QA, reward normalized answer entropy from a tool-enabled solver that does not receive the intended source paper.
Quantitative structure	For calculation-heavy QA, require mathematical notation, explicit quantities and units, and multi-step derivation.

Among these components, difficulty is the key signal for selecting challenging short-form questions. We instantiate its scorer with GPT-5.5 using web search and code, matching the evaluation access policy. Given q, we draw K responses $\hat{a}_1, \ldots, \hat{a}_K$ from oracle $\mathcal{E}_\phi$, canonicalize them into answer classes $\mathcal{C}(q)$, and let n_c be the number assigned to class c. With $\hat{p}(c \mid q) = n_c/K$, we define the normalized empirical entropy

$$\widetilde{\mathcal{H}}(q) = -\frac{1}{\log_2 K} \sum_{c \in \mathcal{C}(q)} \hat{p}(c \mid q) \log_2 \hat{p}(c \mid q), \qquad 0 \le \widetilde{\mathcal{H}}(q) \le 1.$$

The difficulty reward is $R_{\text{diff}}(q) = \widetilde{\mathcal{H}}(q)$; throughout the paper, entropy thresholds and reported entropy changes refer to this normalized quantity. High values indicate disagreement under the no-source-injection policy, while stable convergence receives a value near zero. Because GPT-5.5 also serves as a primary evaluated model, its benchmark score is not an independent test of the selector. Entropy is not a validity criterion: omitted qualifiers can also create uncertainty. We therefore reject or minimally rewrite defects found by a source-blind audit for materially different answers satisfying every explicit constraint; Section 6 reports prevalence and the controlled repair test.

2.3 Synthesizing Labels and Training the Verifier

For each reasoning step r_t and source context C, a committee of K frontier models independently proposes a binary factual-consistency label $V(r_t, C) \in \{0, 1\}$. Unanimity terminates adjudication. Otherwise, an analyst maps claims to source spans, the committee cross-examines its initial decisions, and three models rotate through Prosecutor, Defense, and Arbiter roles. The majority Arbiter vote sets the final label, and a cascading rule assigns label zero to every later step after the first incorrect step.

The adjudicated tuples form supervised examples for the verifier. We train it by supervised fine-tuning (SFT) under a fixed binary output format and freeze it during generator optimization. At generation time, any paragraph labeled inconsistent with its source context activates the factual gate and sets the trajectory reward to zero. Appendix A.2 gives the SFT objective, and Appendix A gives the complete label-adjudication procedure.

3 Answer Annotation

3.1 MAS Annotation

For question q and source paper C, the MAS first filters invalid or leaky rollouts, applies the source-blind ambiguity adjudication, and deduplicates survivors by normalized hash. Three frontier models then produce tool-assisted answers, which are tolerance-clustered, cross-critiqued, and revised. A Latin-square rotation assigns each model once to Prosecutor, Defense, and Arbiter. The system accepts reference answer A^* and provenance status S only when a majority of Arbiters vote "Correct" and the winning cluster agrees numerically with the candidate answer. We do not treat this consensus as a formal correctness guarantee. Appendix A gives the stage-by-stage procedure.

3.2 Human Alignment

To verify the quality of the MAS annotations, professional annotators assessed 82 valid generated question–reference-answer pairs against their source papers for answer correctness, question relevance, and textual support.

Table 3: Human evaluation results.

Criterion	Primary rating	Intermediate rating	Residual rating
Correctness	Correct: 84.1%	Partially correct: 11.0%	Incorrect: 4.9%
Relevance	Relevant: 98.8%	—	Irrelevant: 1.2%
Support	Fully supported: 96.3%	Partially supported: 3.7%	—

As shown in Table 3, 84.1% of the reference answers were fully correct, 11.0% partially correct, and 4.9% incorrect; 98.8% of the questions were relevant to the source material, and 96.3% of the pairs were fully supported by explicit source evidence. Importantly, the 15.9% not rated fully correct should not be interpreted as a 15.9% ambiguity rate.

On a separate annotation axis, 89.0% admitted a single clear answer and 1.2% received the literal `Ambiguous` label. Appendix A.5 gives the complete error taxonomy; the source-blind Fact500 audit appears in Section 6.

Table 4: Complete 500-question open-ended evaluation. Cells report mean DeepEval score and direct-bootstrap 95% CI.

Model	Presentation	Consistency	Citation association
GPT-5.5	80.2 [79.7, 80.7]	78.4 [77.6, 79.2]	**50.6** [48.6, 52.5]
Gemini-2.5-Pro	**81.9** [81.4, 82.4]	79.5 [78.7, 80.3]	44.9 [42.9, 46.8]
Claude-Sonnet-4.5	81.4 [80.9, 81.9]	**80.7** [79.9, 81.5]	45.8 [43.8, 47.7]

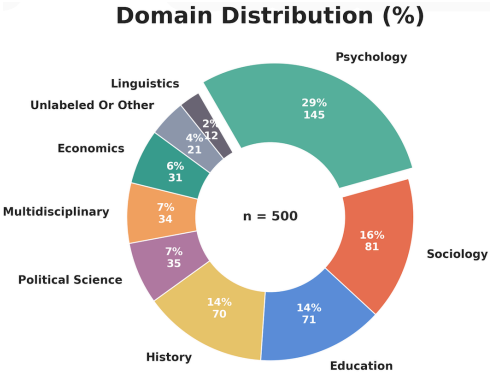


Figure 1: Domain distribution of the 500-item non-STEM QA in U2-Bench.

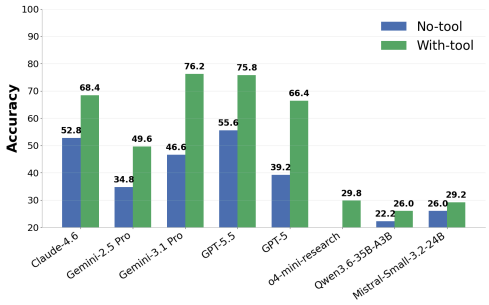


Figure 2: Accuracy of state-of-the-art models under no-tool and with-tool settings on non-STEM QA.

4 U2-BENCH

4.1 SETUP

Training Dataset We construct our training dataset using a dual-pipeline strategy to ensure broad disciplinary coverage. We sample seed papers via the OpenAlex API[1]. Seeds are strictly filtered for quality (English-only, valid abstracts, minimum citations, and STEM denylists) and balanced using per-field quotas. For context formulation, we extract and hydrate each seed's references, sampling 5–15 related papers to concatenate with the seed in the final prompt. For calculation-heavy STEM questions, we start from seed arXiv IDs (Clement et al., 2019) and use the Semantic Scholar API[2] to gather outgoing references and incoming citations. After deduplication and metadata filtering, the prompt combines the seed paper with 5–15 highly cited related papers.

Models We employ gpt-oss-20b (Agarwal et al., 2025) as the model generator. Our difficulty scoring uses GPT-5.5 with web search and coding. We evaluate frontier API models and local open-source models under a unified harness. Frontier models include GPT-5.5 (OpenAI, 2026), GPT-5 (OpenAI, 2025b), Claude-Sonnet-4.5/4.6 (Anthropic, 2025; 2026), Gemini-2.5-Pro/3.1-Pro (Google DeepMind, 2025; 2026), and a deep-research variant (o4-mini-deep-research (OpenAI, 2025a)). Exact versions differ by track and are reported with each result. Open-source local models include Qwen3.6-35B-A3B (Qwen Team, 2026) and Mistral-Small-3.2-24B (Mistral AI, 2025). We evaluate each model in no-tool and with-tool (code+search) settings, except deep-research models that are inherently tool-enabled.

Implementation Details For short-form QA, we generate $K = 8$ stochastic oracle responses per question, canonicalize them into answer classes, and compute normalized entropy $\widetilde{\mathcal{H}}$. Filtering at $\widetilde{\mathcal{H}} \geq 0.7$ yields the fixed 500-question Fact500 evaluation set, which is a deliberately difficulty-selected stress test rather than a natural-distribution estimate. The separate 500-question open-ended set is not selected by answer entropy. Fact500 model scores are computed on this fixed evaluation

[1] https://developers.openalex.org/api-reference/introduction
[2] https://www.semanticscholar.org/product/api

set, and Section 6 reports a separate source-blind validity analysis of the same questions. Generator training uses VeRL (Sheng et al., 2025) on two nodes with eight H200 GPUs each.

4.2 Open-Ended Questions

Evaluation We evaluate all 500 open-ended questions using the same report-generation and DeepEval protocol (Wang et al., 2026a). Each agent must synthesize a comprehensive report and ground its claims in cited sources rather than return a short answer. We assess three complementary dimensions: presentation measures the organization and readability of the report, consistency measures its factual and logical coherence, and citation association measures whether individual claims are linked to supporting literature. We report direct-bootstrap 95% confidence intervals over the full 500-question set for each criterion.

Main Results Across the complete set, all three agents produce well-structured and internally coherent reports: presentation scores range from 80.2 to 81.9 and consistency scores from 78.4 to 80.7 (Table 4). Gemini-2.5-Pro obtains the highest presentation score (81.9), Claude-Sonnet-4.5 the highest consistency score (80.7), and GPT-5.5 the highest citation-association score (50.6). However, citation association remains only 44.9–50.6, trailing presentation by 29.6–36.7 points for every evaluated agent. Thus, systems can generate fluent, well-organized reports while still failing to connect individual claims to the correct supporting literature. We use the confidence intervals to characterize criterion-level uncertainty, not to claim significance for small between-model differences. Precise claim-to-source mapping, rather than report fluency alone, remains the central open-ended failure mode. Table 6 gives an example.

4.3 Fact-Heavy QA (Non-STEM Domains)

Dataset Composition We introduce a 500-item fact-heavy QA benchmark drawn from non-STEM domains. As illustrated in Figure 1, the largest concentrations are Psychology (29%, $n = 145$), Sociology (16%, $n = 81$), Education (14%, $n = 71$), and History (14%, $n = 70$), with the remainder spanning Political Science, Economics, Linguistics, and multidisciplinary topics. Questions require precise retrieval and reasoning over historical or domain-specific evidence. Table 7 gives an example.

Main Results. On the fixed Fact500 evaluation set (Figure 2), tools raise Gemini-3.1-Pro from 46.6% to 76.2% and GPT-5.5 from 55.6% to 75.8%, while the two smaller open-weight models remain below 30%. The complete model comparison appears in Appendix C.3; Section 6 audits validity on the same 500 questions.

4.4 Calculation-Heavy STEM Questions

Construction and Evaluation. To evaluate generalizability beyond fact-heavy non-STEM QA, we use the same generation framework to construct a fixed 500-question calculation-heavy STEM evaluation set. We use gpt-oss-20b as the generator and progressively strengthen the tool-enabled difficulty oracle from GPT-4o-mini to GPT-5-mini and then GPT-5.5 (OpenAI, 2025b; 2026). The generator receives source and citation-graph context, while the oracle receives no injected source paper and retains general web search and code execution.

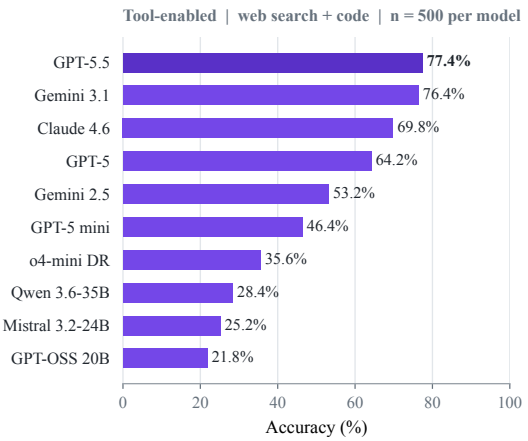


Figure 3: Tool-enabled Calc500 accuracy.

Main Results. Across Calc500, tool-enabled accuracy ranges from 21.8% to 77.4% (Figure 3). GPT-5.5 achieves the highest accuracy at 77.4%, narrowly exceeding Gemini-3.1-Pro at 76.4%, followed by Claude-Sonnet-4.6 at 69.8% and GPT-5 at 64.2%. Performance drops substantially for smaller models: GPT-5-mini reaches 46.4%,

while all evaluated open-weight models remain below 30%. Because every system receives the same web-search and code-execution access, the 55.6-point spread indicates that tool availability alone is insufficient; models must also identify the correct assumptions, construct multi-step derivations, and verify numerical results. Calc500 therefore remains discriminative even among tool-enabled frontier systems.

5 Further Analysis

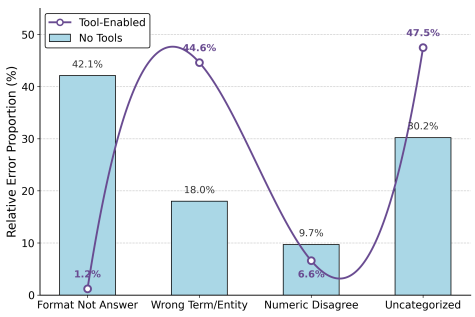


Figure 4: Relative error proportions.

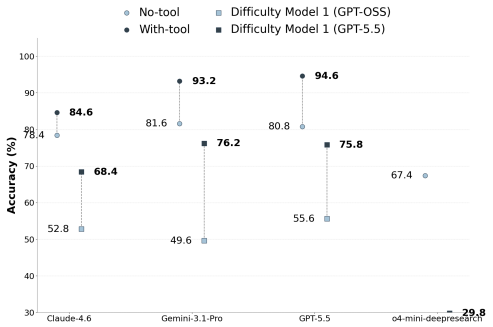


Figure 5: Impact of different difficulty models.

5.1 Failure Analysis

We evaluate GPT-5.5 on the 500-item fact-heavy QA track under *no-tool* and *tool-enabled* settings. We track end-to-end judge accuracy and paired transitions to measure tool-induced recoveries ($N_{\mathrm{I}\to\mathrm{C}}$:= from incorrect to correct) versus regressions ($N_{\mathrm{C}\to\mathrm{I}}$:= from correct to incorrect). To understand failure modes, we classify incorrect outputs into five categories: *format-not-answer*, *wrong term/entity*, *numeric disagreement*, *specificity mismatch*, and *incomplete/partial*.

Quantitative Findings. Tool access improves accuracy from 55.6% to 75.8%, a $+20.2$ point gain. Recoveries strongly outweigh regressions ($N_{\mathrm{I}\to\mathrm{C}} = 135$ vs. $N_{\mathrm{C}\to\mathrm{I}} = 16$).

Error Shifts and Diagnostics. As shown in Figure 4, without tools, errors are predominantly procedural (*format-not-answer* at 42.1%), often exhibiting generation pathologies like excessive length (mean $\approx$ 891 vs. 508 characters) and markup leakage. With tools enabled, output lengths stabilize ($\approx$ 415 characters) and formatting failures nearly vanish (1.2%). Consequently, residual errors shift toward epistemic bottlenecks: *wrong term/entity* failures surge in relative proportion from 18.0% to 44.6%. Aggregate accuracy obscures tool mechanics: tools primarily resolve procedural failures rather than semantic ones, leaving precision as the principal challenge.

5.2 Ablation Analysis

Difficulty Model. Figure 5 shows that oracle choice materially changes benchmark difficulty. With the weaker gpt-oss-120b oracle (circles), no-tool accuracies hover near 80% for several models, creating a ceiling that compresses observed tool gains. Selection with GPT-5.5 (squares) lowers scores into the 50% range and exposes larger tool/no-tool differences. This ablation demonstrates oracle dependence and motivates periodic regeneration under documented benchmark versions; it does not establish an oracle-independent measure of inherent capability. Complementarily, under a fixed source pool and screening policy, 200 generator-training

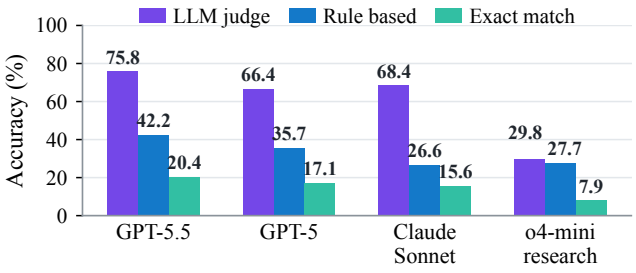


Figure 6: Accuracy across three evaluation metrics.

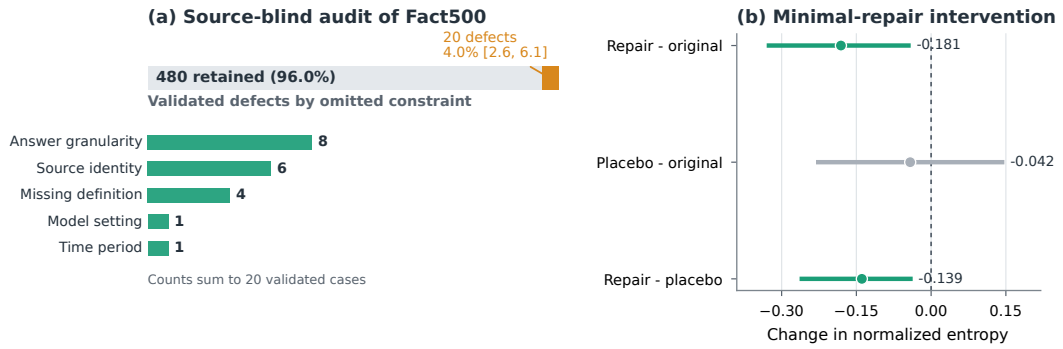


Figure 7: Source-blind validity analysis on Fact500. (a) The complete 500-question audit validates 20 open-world defects (4.0%, Wilson 95% CI: 2.6–6.1); bars decompose the validated cases by omitted constraint. (b) Paired changes in normalized entropy for the 20 validated defects; points show means and whiskers show bootstrap 95% CIs.

steps raise usable yield from 3.6% to 28.4%, well-posedness from 44% to 83%, and mean normalized entropy from 0.71 to 0.93 (Appendix C.1).

Evaluation Metrics. Figure 6 shows substantial sensitivity to the evaluation rule. GPT-5.5 receives 75.8% under the LLM judge, 42.2% under the rule-based metric, and 20.4% under exact match. Rigid matching misses semantically equivalent responses, but LLM judging introduces its own model sensitivity. We therefore report the judge and prompt explicitly and avoid interpreting close score differences as evaluator-independent.

6 Benchmark Validity Analysis

Source-Blind Audit. We audit all 500 Fact500 questions without exposing either the intended paper or the MAS reference answer. A search-enabled adversary proposes independently supported conflicting answers, and a separate adjudicator validates a defect only when both answers satisfy every explicit constraint and differ because of an omitted qualifier. The audit identifies 20 defects (4.0%, Wilson 95% CI: 2.6–6.1), while no defect is validated for the remaining 480 questions. Most defects involve answer granularity (8/20), source identity (6/20), or missing definitions (4/20), with one model-setting and one temporal omission (Figure 7a).

Controlled Minimal Repair. For each validated defect, we compare the original question, a minimal repair that adds only the missing qualifier, and a placebo paraphrase that adds no constraint, using the same tool-enabled difficulty oracle ($K = 4$ samples per variant). Minimal repair lowers normalized entropy by 0.181 relative to the original (95% CI: 0.041–0.330) and by 0.139 relative to placebo (95% CI: 0.037–0.264), whereas the placebo–original interval includes zero (Figure 7b).

Implications. The intervention confirms the proposed mechanism: omitted qualifiers can inflate entropy. However, the complete audit shows that this issue affects a small minority of Fact500 rather than explaining the benchmark-wide difficulty signal. We therefore retain normalized entropy for difficulty selection and pair it with a source-blind reject-or-rewrite validity layer. Reported Fact500 model scores are computed on the fixed evaluation set defined in Section 4.

7 U2-Train

Setting. `U2-Train` is defined as a fully self-evolving variant of the question-generation framework. Let M_0 denote the base model. One copy is trained as the question generator M_{gen}. During rollout, a separate copy of the same base model, M_{diff}, answers each generated question multiple times and

supplies the answer distribution used by the difficulty reward. A fresh solver M_{solve}, also initialized from M_0, is then trained on the generated data after validation. Thus, the question generator, difficulty model, and target solver are separate model instances derived from the same pretrained initialization rather than one model instance trained sequentially across all three roles.

Self-evolution objective. In contrast to `U2-Bench`, which uses a stronger external difficulty model to generate frontier-level evaluations, `U2-Train` targets the base model's own capability gaps. The generator proposes complex questions, the same-base difficulty model identifies those it cannot answer reliably, and the resulting validated questions provide supervision for the same-base solver. External benchmarks spanning tool use, multi-source factual synthesis, and expert-level knowledge are used to test transfer beyond the generated question distribution.

Downstream transfer. Our transfer experiment instantiates `U2-Train` with M_{gen}, M_{diff}, and a fresh M_{solve} initialized from the same GPT-OSS-20B base model. The same-base pipeline generates and filters a joint fact-heavy and calculation-heavy set; we train the fresh solver with GSPO (Zheng et al., 2025) and evaluate it on three external benchmarks under identical settings. Table 5 summarizes the resulting scores: 59.3–64.1 on GAIA, 76.2–79.5 on FRAMES, and 16.6–20.2 on HLE.

Table 5: U2-Train transfer results.

Benchmark	Base	`U2-Train`	Gain
GAIA	59.3	64.1	**+4.8**
FRAMES	76.2	79.5	**+3.3**
HLE	16.6	20.2	**+3.6**

8 Related Work

Challenging Benchmarks As LLMs rapidly advance, evaluation has shifted toward complex, long-horizon, and expert-level benchmarks. Early efforts like GAIA (Mialon et al., 2023) tested general tool use but were limited in scale and relied on costly human annotation. Recent expert-curated datasets, such as HLE (Phan et al., 2025) and FrontierScience (Wang et al., 2026b), offer rigorous multi-domain and scientific evaluation but incur very high construction costs. Concurrently, deep research and web-navigation benchmarks like DeepResearch (Du et al., 2025), BrowseComp (Wei et al., 2025), and DeepSearchQA (Gupta et al., 2026) have emerged. However, these remain predominantly static and vulnerable to data contamination. While LiveResearch (Wang et al., 2026a) introduces dynamic evaluation, it still requires high continuous maintenance.

Data Synthesis The prohibitive costs of human annotation have driven a surge in automated data synthesis using LLMs. Recent surveys highlight how LLMs are transforming data pipelines through prompt-based generation, retrieval-augmented synthesis, and iterative self-refinement (Long et al., 2024; Nădăş et al., 2025). While traditionally used for instruction tuning and augmenting low-resource tasks, synthesis is increasingly applied to dynamic benchmark generation to combat test-set leakage and rapid model saturation (Zhou et al., 2026; Shashidhar et al., 2025). Despite challenges like factual inaccuracies or lack of diversity, modern pipelines mitigate these issues through automated verification and iterative refinement (Long et al., 2024).

9 Conclusion

We introduced `U2-Bench`, an RL-generated, source-grounded stress test with 500 questions in each of three tracks: open-ended reports, fact-heavy QA, and calculation-heavy STEM questions. On Fact500, web search and code improve GPT-5.5 accuracy by 20.2 points, and the source-blind audit validates 20 open-world defects among 500 questions. For those defects, minimal repairs reduce normalized oracle entropy relative to both the original questions and placebo paraphrases, whereas the placebo–original interval includes zero. On the open-ended track, citation association remains substantially weaker than presentation and consistency, identifying claim-to-source mapping as the central failure mode. Calc500 remains discriminative among tool-enabled systems, with accuracy ranging from 21.8% to 77.4%. Finally, `U2-Train` improves the same-base solver by 4.8, 3.3, and 3.6 points on GAIA, FRAMES, and HLE, respectively. Together, these results show that source-grounded question generation can support both renewable evaluation and positive downstream transfer.

AI USE STATEMENT

Generative AI models are core experimental components of this work: they generate questions, provide difficulty signals, support verification and answer annotation, and serve as evaluated systems. Generative AI tools also assisted with literature retrieval, code and artifact inspection, manuscript organization, and language editing. The authors reviewed these uses and take full responsibility for the paper's claims, analysis, and presentation.

ETHICS STATEMENT

Automatically generated questions can inherit factual errors, ambiguities, and biases from source materials and model judgments. We mitigate these risks through source grounding, independent validity checks, and documented benchmark versions. The benchmark is intended for model evaluation and training research rather than high-stakes decision making.

REPRODUCIBILITY STATEMENT

The paper specifies the generation, difficulty-selection, verification, and evaluation procedures, and the appendix records additional implementation and validity-analysis details. Quantitative claims are limited to the reported experimental settings; the limitations identify the additional controls required for causal or robustness conclusions.

REFERENCES

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. GPT-OSS-120B & GPT-OSS-20B model card. *arXiv preprint arXiv:2508.10925*, 2025.

Anthropic. Introducing Claude Sonnet 4.5, September 2025. URL https://www.anthropic.com/news/claude-sonnet-4-5. Released: September 29, 2025. Accessed: 2026-09-26.

Anthropic. Introducing Claude Sonnet 4.6, February 2026. URL https://www.anthropic.com/news/claude-sonnet-4-6. Released: February 17, 2026. Accessed: 2026-09-26.

Simin Chen, Yiming Chen, Zexin Li, Yifan Jiang, Zhongwei Wan, Yixin He, Dezhi Ran, Tianle Gu, Haizhou Li, Tao Xie, and Baishakhi Ray. Benchmarking large language models under data contamination: A survey from static to dynamic evaluation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 10080–10098, Suzhou, China, November 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.511. URL https://aclanthology.org/2025.emnlp-main.511/.

Colin B Clement, Matthew Bierbaum, Kevin P O'Keeffe, and Alexander A Alemi. On the use of arXiv as a dataset. *arXiv preprint arXiv:1905.00075*, 2019.

Jasper Dekoninck, Nikola Jovanović, Tim Gehrunger, Kári Rögnvalddson, Ivo Petrov, Chenhao Sun, and Martin Vechev. Beyond benchmarks: MathArena as an evaluation platform for mathematics with LLMs, 2026. URL https://arxiv.org/abs/2605.00674.

Mingxuan Du, Benfeng Xu, Chiwei Zhu, Xiaorui Wang, and Zhendong Mao. DeepResearch Bench: A comprehensive benchmark for deep research agents. *arXiv preprint arXiv:2506.11763*, 2025.

Google DeepMind. Gemini 2.5 Pro model card, June 2025. URL https://deepmind.google/models/model-cards/gemini-2-5-pro. Published: June 27, 2025. Accessed: 2026-09-26.

Google DeepMind. Gemini 3.1 Pro model card, February 2026. URL https://deepmind.google/models/model-cards/gemini-3-1-pro. Published: February 19, 2026. Accessed: 2026-09-26.

Nikita Gupta, Riju Chatterjee, Lukas Haas, Connie Tao, Andrew Wang, Chang Liu, Hidekazu Oiwa, Elena Gribovskaya, Jan Ackermann, John Blitzer, et al. DeepSearchQA: Bridging the comprehensiveness gap for deep research agents. *arXiv preprint arXiv:2601.20975*, 2026.

Vincent Hsiao, Morgan Fine-Morris, Mark Roberts, Leslie N Smith, and Laura M. Hiatt. A critical assessment of LLMs for solving multi-step problems: Preliminary results. In *AAAI 2025 Workshop LM4Plan*, 2025. URL https://openreview.net/forum?id=kFrqoVtMIy.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*, 2024.

Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? *arXiv preprint arXiv:2310.06770*, 2023.

Satyapriya Krishna, Kalpesh Krishna, Anhad Mohananey, Steven Schwarcz, Adam Stambler, Shyam Upadhyay, and Manaal Faruqui. Fact, fetch, and reason: A unified evaluation of retrieval-augmented generation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 4745–4759, Albuquerque, New Mexico, April 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.naacl-long.243. URL https://aclanthology.org/2025.naacl-long.243/.

Lin Long, Rui Wang, Ruixuan Xiao, Junbo Zhao, Xiao Ding, Gang Chen, and Haobo Wang. On LLMs-driven synthetic data generation, curation, and evaluation: A survey. In *Findings of the Association for Computational Linguistics: ACL 2024*, pp. 11065–11082, 2024.

Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. GAIA: A benchmark for general AI assistants. *arXiv preprint arXiv:2311.12983*, 2023.

Mistral AI. Mistral-small-3.2-24b-instruct-2506. https://huggingface.co/mistralai/Mistral-Small-3.2-24B-Instruct-2506, 2025. Accessed: 2026-05-05.

Mihai Nădăş, Laura Dioşan, and Andreea Tomescu. Synthetic data generation using large language models: Advances in text and code. *IEEE Access*, 13:134615–134633, 2025. ISSN 2169-3536. doi: 10.1109/ACCESS.2025.3589503. URL https://doi.org/10.1109/ACCESS.2025.3589503.

OpenAI. Introducing Deep Research. https://openai.com/index/introducing-deep-research/, 2025a. Accessed: 2026-05-05.

OpenAI. GPT-5 system card, August 2025b. URL https://openai.com/index/gpt-5-system-card/. Published: August 7, 2025. Accessed: 2026-09-26.

OpenAI. Introducing GPT-5.5, April 2026. URL https://openai.com/index/introducing-gpt-5-5/. Released: April 23, 2026. Accessed: 2026-09-26.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Chen Bo Calvin Zhang, Mohamed Shaaban, John Ling, Sean Shi, et al. Humanity's last exam. *arXiv preprint arXiv:2501.14249*, 2025.

Changle Qu, Sunhao Dai, Xiaochi Wei, Hengyi Cai, Shuaiqiang Wang, Dawei Yin, Jun Xu, and Ji-rong Wen. Tool learning with large language models: A survey. *Frontiers of Computer Science*, 19(8):198343, January 2025. ISSN 2095-2236. doi: 10.1007/s11704-024-40678-2. URL https://doi.org/10.1007/s11704-024-40678-2.

Qwen Team. Qwen3.6-35B-A3B: Agentic coding power, now open to all, April 2026. URL https://qwen.ai/blog?id=qwen3.6-35b-a3b.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, et al. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.

Sumuk Shashidhar, Clémentine Fourrier, Alina Lozovskia, Thomas Wolf, Gokhan Tur, and Dilek Hakkani-Tür. YourBench: Easy custom evaluation sets for everyone. *arXiv preprint arXiv:2504.01833*, 2025.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. HybridFlow: A flexible and efficient RLHF framework. In *Proceedings of the Twentieth European Conference on Computer Systems*, EuroSys '25, pp. 1279–1297. ACM, 2025. doi: 10.1145/3689031.3696075. URL https://doi.org/10.1145/3689031.3696075.

Zhengliang Shi, Shen Gao, Lingyong Yan, Yue Feng, Xiuyi Chen, Zhumin Chen, Dawei Yin, Suzan Verberne, and Zhaochun Ren. Tool learning in the wild: Empowering language models as automatic tool agents. In *Proceedings of the ACM Web Conference 2025*, pp. 2222–2237. ACM, 2025. doi: 10.1145/3696410.3714825. URL https://doi.org/10.1145/3696410.3714825.

Yifan Sun, Han Wang, Dongbai Li, Gang Wang, and Huan Zhang. The emperor's new clothes in benchmarking? a rigorous examination of mitigation strategies for LLM benchmark data contamination. *arXiv preprint arXiv:2503.16402*, 2025.

Chaojie Wang, Yanchen Deng, Zhiyi Lyu, Liang Zeng, Jujie He, Shuicheng Yan, and Bo An. Q*: Improving multi-step reasoning for LLMs with deliberative planning, 2024. URL https://arxiv.org/abs/2406.14283.

Jiayu Wang, Yifei Ming, Riya Dulepet, Qinglin Chen, Austin Xu, Zixuan Ke, Frederic Sala, Aws Albarghouthi, Caiming Xiong, and Shafiq Joty. LiveResearchBench: A live benchmark for user-centric deep research in the wild. In *International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=ghwbZ3uhEd.

Miles Wang, Robi Lin, Kat Hu, Joy Jiao, Neil Chowdhury, Ethan Chang, and Tejal Patwardhan. FrontierScience: Evaluating AI's ability to perform expert-level scientific tasks. *arXiv preprint arXiv:2601.21165*, 2026b.

Jason Wei, Zhiqing Sun, Spencer Papay, Scott McKinney, Jeffrey Han, Isa Fulford, Hyung Won Chung, Alex Tachard Passos, William Fedus, and Amelia Glaese. BrowseComp: A simple yet challenging benchmark for browsing agents. *arXiv preprint arXiv:2504.12516*, 2025.

Bin Xu, Yu Bai, Huashan Sun, Yiguan Lin, Siming Liu, Xinyue Liang, Yaolin Li, Zhuangzhi Dong, Jingren Zhang, Yufan Deng, et al. EduBench: A comprehensive benchmarking dataset for evaluating large language models in diverse educational scenarios. *arXiv preprint arXiv:2505.16160*, 2025a.

Cheng Xu, Nan Yan, Shuhao Guan, Changhong Jin, Yuke Mei, Yibing Guo, and Tahar Kechadi. DCR: Quantifying data contamination in LLM evaluation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 23002–23020, Suzhou, China, November 2025b. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.1173. URL https://aclanthology.org/2025.emnlp-main.1173/.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. Group sequence policy optimization, 2025. URL https://arxiv.org/abs/2507.18071.

Yajie Zhou, Jiajun Ruan, Eric S. Wang, Sadjad Fouladi, Francis Y. Yan, Kevin Hsieh, and Zaoxing Liu. NetArena: Dynamic benchmarks for AI agents in network automation, 2026. URL https://arxiv.org/abs/2506.03231. Version 2, revised March 13, 2026.

A Detailed Method Definitions

A.1 Reward Definitions

Paragraph-based factual gate. The frozen SFT-trained verifier evaluates every reasoning paragraph against the supplied source context using the binary label format defined in Section 2.3. Any paragraph labeled inaccurate or unsupported sets the trajectory reward to zero; the remaining reward components are evaluated only after this shared gate is passed.

Format and selection. The format reward requires the prescribed reasoning, question, and source-selection fields to be present and machine-parseable. The selection reward checks that the chosen papers or sections form a valid, appropriately sized subset and include the seed source. Open-ended generation selects papers, whereas deterministic short-form generation selects sections.

Lexical self-containment. This reward penalizes prompt-local references such as *in this paper*, *the above section*, or undefined local labels. It requires the question to stand on its own linguistically, but does not require source-specific evidence to be recoverable without retrieval.

Coverage. For open-ended questions, the coverage reward favors technical depth, synthesis across the selected papers, and source-specific concepts over generic topical overlap. It is not used for deterministic short-form questions.

Difficulty. For fact-heavy and calculation-heavy QA, a tool-enabled solver receives the generated question without an injected copy of the intended source paper. We canonicalize multiple sampled answers and reward high normalized empirical entropy $\widetilde{\mathcal{H}}$ while penalizing stable convergence. Because uncertainty may also arise from omitted constraints, difficulty is not treated as validity; source-blind adjudication separately rejects or minimally repairs questions admitting materially different valid answers (Section 6).

Quantitative structure. For calculation-heavy QA, this reward requires explicit mathematical structure, quantities or units, and a multi-step computation or derivation. It complements the factual gate: satisfying the expected quantitative form cannot rescue an unsupported trajectory.

A.2 Optimization Objectives

Generator optimization. For context C, the behavior policy $\pi_{\theta_{\text{old}}}$ samples N trajectories $\{y_i\}_{i=1}^{N}$. The scalar reward is

$$R(y) = G_{\text{fact}}(y) \sum_{j \in \mathcal{J}(y)} w_j R_j(y),$$

where $G_{\text{fact}}(y) \in \{0, 1\}$ is the paragraph-level factual gate, $\mathcal{J}(y)$ is the set of active components, and w_j is a fixed training coefficient. GRPO maximizes

$$\begin{aligned} \mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}\Bigg[\frac{1}{N} \sum_{i=1}^{N} \frac{1}{|y_i|} \sum_{t=1}^{|y_i|} \min \Big(\rho_{i,t}(\theta) A_i, \\ \text{clip}(\rho_{i,t}(\theta), 1-\epsilon, 1+\epsilon)\, A_i \Big) - \beta\, \mathbb{D}_{\text{KL}}[\pi_\theta \,\|\, \pi_{\text{ref}}] \Bigg], \end{aligned} \tag{2}$$

where $\rho_{i,t}(\theta) = \pi_\theta(y_{i,t} \mid C, y_{i,<t}) / \pi_{\theta_{\text{old}}}(y_{i,t} \mid C, y_{i,<t})$, ϵ is the clipping radius, and β is the KL coefficient. For $R_i = R(y_i)$, the group-relative advantage is $A_i = (R_i - \bar{R}_C)/(s_C + \delta)$, where $\bar{R}_C = N^{-1} \sum_j R_j$, s_C is the within-group standard deviation, and $\delta > 0$ is a numerical stabilizer. We set $A_i = 0$ when all group rewards are equal.

Verifier SFT. The adjudicated tuples define $\mathcal{D}_V = \{(C_i, r_i, v_i)\}_{i=1}^{N_V}$. With the fixed binary output format, the verifier is trained by next-token cross-entropy,

$$\mathcal{L}_V(\psi) = -\frac{1}{N_V} \sum_{i=1}^{N_V} \log p_\psi(v_i \mid C_i, r_i).$$

The trained verifier is frozen before generator optimization and supplies the paragraph-level factual gate described above.

A.3 Step-Wise Verifier Labeling

For a reasoning step r_t and source context C, the verifier-labeling MAS assigns $V(r_t, C) \in \{0, 1\}$ through four stages.

Stage 1: Independent Evaluation. A committee of K frontier models independently evaluates r_t against C, casting votes $v_k^{(1)} \in \{0, 1\}$. A unanimous vote terminates the process with the agreed label.

Stage 2: Evidence Extraction. If the initial vote is disputed, an analyst extracts structured claims $E = \{(c_i, src_i)\}$ and maps every claim c_i to a specific source span $src_i \in C$.

Stage 3: Cross-Examination. The committee re-evaluates its decisions conditioned on E and the peer reasoning, producing updated votes $v_k^{(3)}$.

Stage 4: Rotated Tribunal. If disagreement remains, three models rotate through Prosecutor, Defense, and Arbiter roles over three rounds, so that each model occupies every role once. The final label is the majority vote of the three Arbiters. A cascading rule invalidates all subsequent reasoning steps after the first incorrect step.

A.4 Reference-Answer Annotation

For each generated question q and source context C, the answer-annotation MAS assigns a reference answer A^* and provenance status S through four stages.

Stage 1: Filtering, Validity, and Deduplication. Rollouts are filtered for formatting, leakage, and difficulty. A source-blind adversary searches for independently supported conflicting answers, and a separate adjudicator accepts a defect only when both answers satisfy every explicit constraint and differ because of an omitted qualifier. Validated defects are rejected or minimally rewritten; survivors are deduplicated by normalized hash.

Stage 2: Parallel Proposal. Three frontier models independently solve q with code and web-search tools. A tolerance-aware extractor groups formatting and floating-point variants into answer clusters.

Stage 3: Cross-Critique. Each model reviews peer reasoning and answers, flags arithmetic or factual errors, and may issue a revised proposal. The largest resulting numeric cluster becomes the candidate $\hat{a}$.

Stage 4: Aggregation and Rotated Tribunal. The models rotate through Prosecutor, Defense, and Arbiter roles using a three-round Latin square. A valid consensus requires both a majority of `Correct` Arbiter votes and strict numeric agreement between the winning Arbiter cluster and $\hat{a}$.

A.5 Human-Review Error Taxonomy

Reanalysis of the professional annotators' written justifications attributes 4.9% of the reviewed cases to genuine uniqueness defects, 2.4% to component-level all-or-nothing scoring, 2.4% to granularity disagreements, 1.2% to cases without usable justification, and 4.9% to reference-answer errors supported by alternative evidence. Source support alone does not establish answer uniqueness; this source-aware review therefore complements the complete source-blind Fact500 audit in Section 6.

B Concrete examples in U2−Bench

Open-Ended Questions Table 6 shows a concrete example of open-ended question.

Table 6: Concrete Example of Open-Ended Questions

Problem Statement
How does the critical scaling of mutual robustness of magic observed in the quantum Ising chain via quantum Monte Carlo tomography relate to the monotonic resource measures (such as the sum of negative Wigner entries) that quantify magic in stabilizer-based resource theories, and what are the implications of this relationship for the temperature dependence and efficiency of magic-state distillation protocols in fault-tolerant quantum computation?

Fact-Heavy QA (Non-STEM Domains) Table 7 shows a concrete example.

Table 7: Fact-heavy QA question from a non-STEM domain.

Problem Statement
In the 1972 presidential election, white voters' opposition to school busing was strongly associated with symbolic attitudes such as racial intolerance and political conservatism, whereas personal economic grievances had little influence on congressional voting between 1956 and 1976. Which category of attitudes was identified as having a stronger influence on the busing opposition?

Calculation-Heavy STEM Table 8 shows a concrete calculation-heavy question.

Table 8: STEM Questions.

Problem Statement
Consider a two-dimensional two-band metal with inter-band pairing. The system undergoes a quantum normal-to-FFLO transition when the hybridization V reaches a critical value V_c that equals the homogeneous superconducting gap Δ_0. The mismatch of the Fermi wave-vectors of the hybridized bands is given by $$\delta k_F \;=\; \frac{2\,V}{\hbar\,\bar{v}_f},$$ where $\hbar = 1.0545718 \times 10^{-34}$ J·s and $\bar{v}_f$ is the average Fermi velocity. The characteristic wave-vector magnitude of the inhomogeneous superconducting order parameter at the transition equals this mismatch, i.e. $q_c = \delta k_F$. Given $$\Delta_0 = 2.0 \text{ meV}, \qquad \bar{v}_f = 1.0 \times 10^5 \text{ m·s}^{-1},$$ and the conversion $1 \text{ eV} = 1.602 \times 10^{-19}$ J, compute the magnitude of the critical wave-vector q_c in units of nm^{-1}. Provide your answer to two significant figures.

C Additional Controlled Experiments

C.1 Generator Training Dynamics

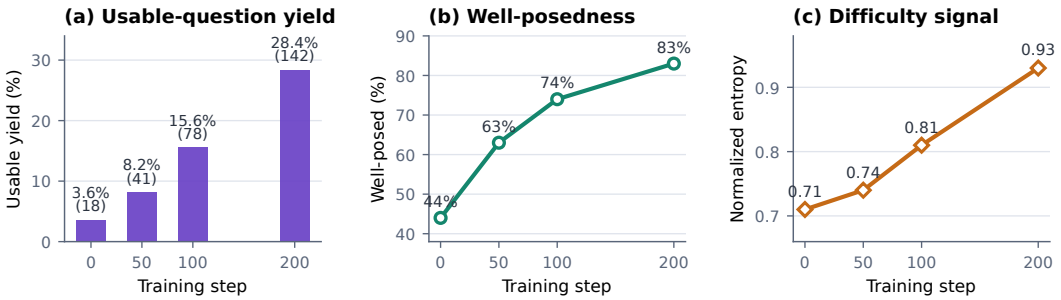


Figure 8: Generator training dynamics under a fixed source pool and screening policy. Each evaluated training step generates 500 candidates and applies the same GPT-5.5 screening policy. (a) Usable-question yield, with counts in parentheses. (b) Fraction classified as well posed. (c) Mean normalized answer entropy.

Before training, the generator yields 18 usable questions from 500 candidates (3.6%). At 50 steps, usable yield rises to 8.2% and well-posedness to 63%, while normalized entropy changes only from 0.71 to 0.74, indicating that the initial gains primarily improve question usability. By step 200, usable yield reaches 28.4% (142/500), well-posedness reaches 83%, and normalized entropy reaches 0.93. The simultaneous rise in well-posedness and entropy suggests that the later difficulty gain is not explained solely by degraded question formulation. Because both screening and measurement follow the same GPT-5.5-aligned protocol, these are within-protocol training dynamics rather than an independent test of generalization; they also do not isolate the difficulty reward from the other reward components.

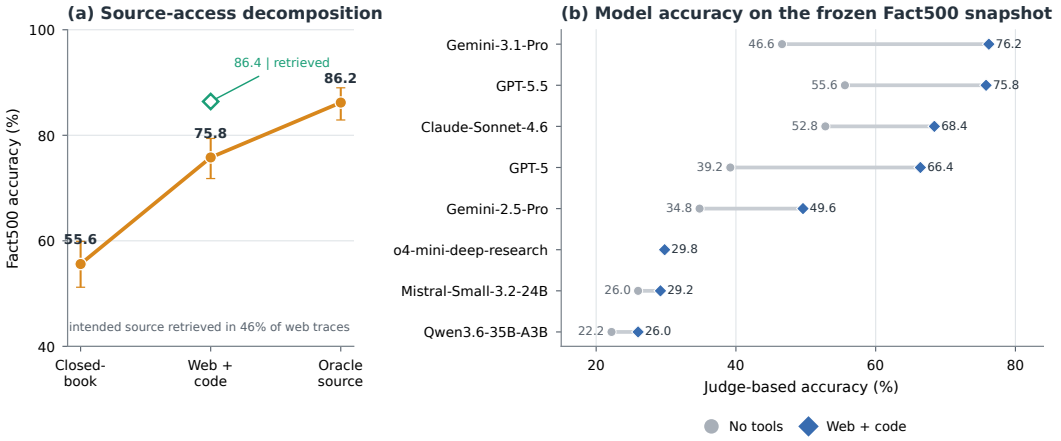


Figure 9: Fact500 access and model results on the fixed evaluation set. (a) Matched GPT-5.5 source-access decomposition over 500 questions: closed-book 55.6% [51.2, 59.9], web search plus code 75.8% [71.8, 79.4], and oracle source 86.2% [82.9, 89.0]. The intended source is retrieved in 46% of web traces, where conditional accuracy is 86.4%. Whiskers are Wilson 95% CIs. (b) Judge-based model accuracy without tools and with web search plus code. o4-mini-deep-research is inherently tool-enabled and therefore has no no-tool point.

C.2 Source-Access Decomposition

The matched protocol distinguishes closed-book capability, tool-mediated retrieval and reasoning, and reasoning with the intended source already available. For the web-and-code condition, source

retrieval is counted only when the trace contains the intended paper or an independently verified equivalent source. Short-form accuracy is scored under the same answer policy in all three conditions.

For Fact500, web search and code improve accuracy by 20.2 points over the closed-book condition, while direct access to the intended source adds a further 10.4 points. The intended paper is retrieved in 46% of web-and-code traces; conditioned on successful retrieval, accuracy rises to 86.4%. Thus, code and general web evidence provide substantial value even without the intended paper, while direct source access closes an additional source-specific gap (Figure 9(a)). We do not report a matched source-access decomposition for the calculation-heavy evaluation set.

C.3 Fact500 Model Evaluation

Figure 9(b) compares the same judge-based accuracy under the no-tool and web-plus-code conditions. Tool access improves every model for which both conditions are available, while the magnitude varies substantially across models.

D Additional Validation Details

D.1 Committee Configuration and Reference Provenance

Table 9: Configuration of the verifier and answer-annotation committees.

Pipeline	Committee	Access	Decision procedure
Step-wise verifier	Claude-Sonnet-4.6, GPT-5.4, Gemini-3.1-Pro	Source context; temperature 0	Independent votes; evidence extraction on disagreement; cross-examination; three-round Latin-square tribunal
Quantitative answers	GPT-5.5, Claude-Sonnet-4.5, Gemini-3.1-Pro	Source context, search, code; temperature 0	Independent proposals; cross-critique; 5% relative-tolerance clustering; rotated tribunal and provenance status

We retain proposals, tool traces, tribunal rulings, and statuses such as `CONSENSUS`, `MAJORITY`, and `JUDGE_OVERRIDE`. Independent Python solutions were also executed in a network- and file-isolated sandbox. Many calculation-heavy questions require source-specific assumptions or domain derivations, so solver disagreement cannot identify the correct answer; this audit is not a symbolic correctness guarantee. Program disagreements and unit/scale inconsistencies should be routed to expert review.

D.2 Hard Source Dependence versus Underspecification

A hard-but-valid item supplies enough identifying constraints to determine a unique supported target, yet requires rare-fact retrieval, cross-paper synthesis, or multi-step derivation. An underspecified item permits materially different supported answers because an entity, comparison set, branch, unit, time period, or answer granularity is omitted. The source-blind adversary and adjudicator operationalize this distinction and record the competing answers and evidence provenance for every rejected or rewritten item.

E Limitations

Several limitations bound our claims. First, entropy depends on oracle calibration and does not certify validity: weak oracles can be uncertain on trivial items, while omitted qualifiers can also increase uncertainty. The source-blind filter reduces but cannot guarantee elimination of all open-world defects. GPT-5.5 supplies the difficulty signal and is also a primary evaluated model, so its benchmark score is not an independent validation of the selector. Second, independently generated benchmark versions are not directly comparable, so documented versioning and anchor evaluations are essential. Third, accepted items remain domain-skewed despite seed-level quotas, limiting aggregate cross-domain conclusions. Fourth, MAS consensus reference answers are not formal correctness proofs, especially

for source-dependent calculations. Calc500 has no independent validity audit, and its reported accuracies do not establish correctness, uniqueness, or computability. Fifth, the full-scale open-ended result covers one writer and one judge, so it does not establish writer or judge invariance. Sixth, the `U2-Train` transfer experiment reports a single training configuration and therefore does not quantify seed-level variability; matched training-data controls are also required to isolate the contribution of self-evolving difficulty selection. Seventh, RL training on two $8\times$H200 nodes is expensive; the efficiency claim concerns marginal expert labor after this one-time cost, not low total compute. Finally, maintaining fresh evaluation material requires continued literature ingestion and does not guarantee permanent non-saturation. `U2-Bench` is therefore a renewable stress test whose versions must be documented for longitudinal comparison.