DR Tulu: Reinforcement Learning with Evolving Rubrics for Deep Research

Rulin Shao[1♡†], **Akari Asai**[2,3♡†], **Shannon Zejiang Shen**[4♡†], **Hamish Ivison**[1,2♡†], **Varsha Kishore**[1,2†], **Jingming Zhuo**[1†], **Xinran Zhao**[3], **Molly Park**[1], **Samuel Finlayson**[1,5], **David Sontag**[4], **Tyler Murray**[2], **Sewon Min**[2,6], **Pradeep Dasigi**[2], **Luca Soldaini**[2], **Faeze Brahman**[2], **Wen-tau Yih**[1], **Tongshuang Wu**[3], **Luke Zettlemoyer**[1], **Yoon Kim**[4], **Hannaneh Hajishirzi**[1,2], **Pang Wei Koh**[1,2]

[1]University of Washington, [2]Allen Institute for AI, [3]Carnegie Mellon University
[4]Massachusetts Institute of Technology, [5]Seattle Children's Hospital, [6]University of California, Berkeley

[♡]Joint first authors, [†]Core contributors. See full author contributions here.

Date: November 18, 2025
Correspondence: rulins@cs.washington.edu, akaria@allenai.org
Code: github.com/rlresearch/dr-tulu
Data & Models: huggingface.co/collections/rl-research/dr-tulu
Blogpost: allenai.org/blog/dr-tulu

Abstract

Deep research models perform multi-step research to produce long-form, well-attributed answers. However, most open deep research models are trained on easily verifiable short-form QA tasks via reinforcement learning with verifiable rewards (RLVR), which does not extend to realistic long-form tasks. We address this with **Reinforcement Learning with Evolving Rubrics (RLER)**, in which we construct and maintain rubrics that co-evolve with the policy model during training; this allows the rubrics to incorporate information that the model has newly explored and to provide discriminative, on-policy feedback. Using RLER, we develop **Deep Research Tulu (DR Tulu-8B)**, the first open model that is directly trained for open-ended, long-form deep research. Across four long-form deep research benchmarks in science, healthcare and general domains, DR Tulu substantially outperforms existing open deep research models, and matches or exceeds proprietary deep research systems, while being significantly smaller and cheaper per query. To facilitate future research, we release all data, models, and code, including our new MCP-based agent infrastructure for deep research systems.

1 Introduction

Deep research (DR) models aim to produce in-depth, well-attributed answers to complex research tasks by planning, searching, and synthesizing information from diverse sources (OpenAI, 2025). Existing open DR models are either training-free, using manually designed prompts with off-the shelf models (Li et al., 2025b,a), or trained indirectly via Reinforcement Learning with Verifiable Rewards (RLVR) on search-intensive, short-form question answering as a proxy for deep research tasks (Jin et al., 2025; Nguyen et al., 2025; Liu et al., 2025). Directly training for deep research is challenging because it is hard to evaluate responses: the desiderata for a good response are often under-specified (Xu et al., 2023; Krishna et al., 2021), so a finite set of static rubrics cannot capture all the dimensions along which a response could be good or

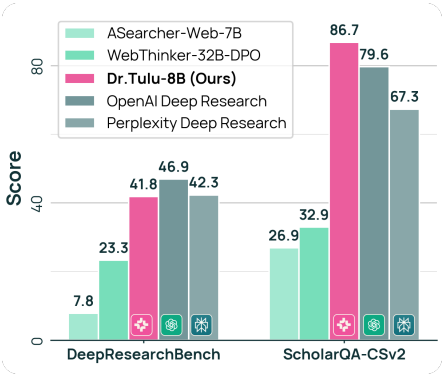

Figure 1 DR Tulu-8B matches or exceeds proprietary deep research models (gray) and outperforms open-source models (green) on deep-research benchmarks.

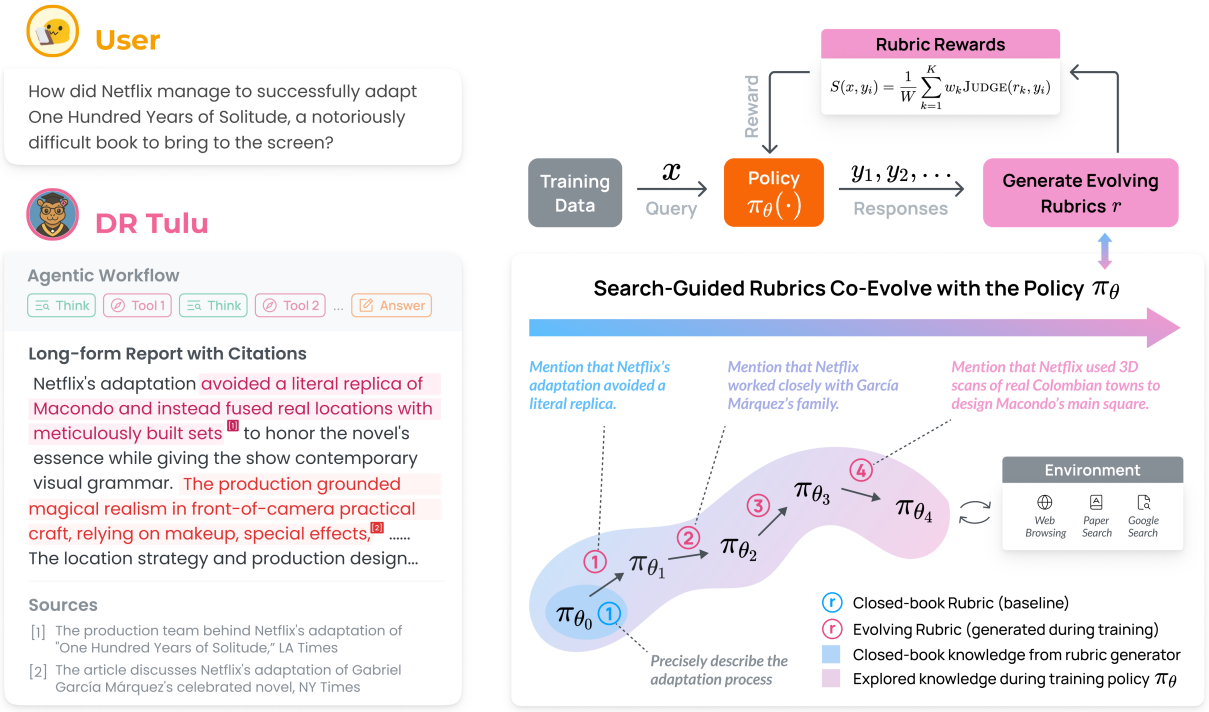

Figure 2 Overview of training a deep research model with reinforcement learning with evolving rubrics (RLER). Left: An example of a question and a long-form response from DR Tulu with citations. Right: We train the policy model on a dynamic set of rubrics that (1) *co-evolve* with the policy update (details in Figure 3) and (2) are grounded on real-world, searched knowledge from the environment. Compared to commonly-used closed-book rubrics generated purely from LM parametric knowledge (blue circle), our evolving rubrics incorporate newly searched information and are continuously tailored to the current policy model's behaviors, better capturing the nuances required for long-form DR tasks.

bad. Moreover, given the knowledge-intensive nature of DR tasks, reliable evaluation requires access to a vast, dynamic corpus of world knowledge, rather than relying solely on the model's parametric knowledge (Gunjal et al., 2025).

In this paper, we introduce **Deep Research Tulu (DR Tulu-8B)**, the first open model that is directly trained for *open-ended, long-form* deep research tasks. To address the challenge of verification in long-form tasks, DR Tulu is first finetuned on high-quality, naturally occurring user data, and then trained via a new method we call **reinforcement learning with evolving rubrics (RLER)**, in which we construct and maintain rubrics that *co-evolve* with the policy model during training. As Figure 2 illustrates, at each training step, we sample several responses and search traces from the model and generate new rubrics that capture and contrast the good and bad points of these responses. This allows us to dynamically incorporate newly explored information into the rubrics and to ensure that they provide on-policy feedback that can discriminate among model responses.

DR Tulu-8B outperforms the strongest open 8-32B models by 8–42 percentage points on four long-form DR benchmarks—AstaBench-ScholarQACS2 (Asai et al., 2024; Bragg et al., 2025), DeepResearchBench (Du et al., 2025), ResearchQA (Yifei et al., 2025), and HealthBench (Arora et al., 2025). In addition, it matches or exceeds proprietary systems (e.g., OpenAI Deep Research, Perplexity Deep Research), while being significantly smaller and cheaper. To assess performance on an out-of-domain, even more challenging real-world expert DR task, we additionally construct GeneticDiseasesQA, a new evaluation dataset annotated by clinicians. This dataset requires models to search for and synthesize supporting evidence to assess the therapeutic eligibility of disease-causing genetic variants. On this benchmark, DR Tulu-8B similarly matches or exceeds existing proprietary DR models, including OpenAI Deep Research.

Our analysis shows that RLER improves the model's ability to produce more comprehensive and in-depth long-form responses with accurate citations, yielding gains of 3–14 points on top of the finetuned model across the four benchmarks. Moreover, DR Tulu learns to select appropriate search tools and arguments

depending on the task, instead of relying on a single hard-coded search tool like in prior work (Gao et al., 2025; Bragg et al., 2025). For instance, on ResearchQA, DR Tulu uses paper search 90% of the time, whereas on DeepResearchBench, whose questions span more diverse, general-domain topics, it relies on web search and browsing about 55% of the time.

We fully release our data, code, and models to support future work on long-form deep research. In particular, we provide a deep research library with MCP search tools (`dr-agent-lib`), together with an evaluation suite, that integrates diverse search and browsing tools and allows new tools to be specified and swapped in with minimal effort. Our training and inference stack supports asynchronous tool calling, making it practical to train and evaluate models on deep-research trajectories with many tool calls. Together, these resources are intended to serve as a simple, extensible foundation for future open deep research agents.

2 Problem Formulation for Deep Research

We consider a *deep research model* to be a language model (LM) equipped with search tools. Each tool takes a query and arguments, returning textual resources that can be cited in the model's answer.

Formally, let $\mathcal{T} = \{T_1, T_2, \ldots\}$ denote the available tools. Each tool T_k takes a query q with optional argument string α and returns an observation $o = T_k(q; \alpha)$. The model's policy π_θ (with parameters θ) operates autoregressively over a sequence of text s, initialized as $s_0 = x$ (the task and system instructions). Concretely, we define the model's action space as { `think`, `tool`, `answer` }, with corresponding protocol tokens:

- `think` (`<think></think>`) uses the LM itself to plan next steps given the current state and information.
- `tool` (`<call_tool></call_tool>`) invokes one of multiple search-related tools. The specific tool is chosen by setting the `name` attribute and tool-specific arguments. *Example:* `<call_tool name="google_search" k="10" lang="en">query</call_tool>`. We append the tool's output, in plain text, to the context for subsequent steps.
- `answer` (`<answer></answer>`) produces the final response and stops. Inside the answer, the LM may enclose claims in `<cite id="SOURCE_ID"></cite>` tags to provide references to supporting source identifiers.

At each step i, the model samples an action and its content or arguments, $(a_i, \zeta_i) \sim \pi_\theta(\cdot \mid s_i)$, where a_i specifies the action type: $a_i =$ `think` for generating reasoning text; $a_i =$ `tool` for calling the corresponding tool T_k with query (q_i, α_i); or $a_i =$ `answer` for producing the final answer. If $a_i \in$ { `answer`, `think` }, the output ζ_i is appended to the context, forming $s_{i+1} = s_i \oplus \langle a_i, \zeta_i \rangle$. If $a_i =$ `tool`, the model executes the tool call, receives $o_i = T_k(q_i; \alpha_i)$, and updates the state as $s_{i+1} = s_i \oplus \langle a_i, \zeta_i, o_i \rangle$. The process continues until $a_\tau =$ `answer`, where ζ_τ contains the final answer. Claims derived from retrieved content should be wrapped in citation tags linking to their supporting evidence.

3 RLER: Reinforcement Learning with Evolving Rubrics

We introduce *Reinforcement Learning with Evolving Rubrics* (*RLER*), our method for training long-form deep-research models using rubrics that are *specific to instances*, *grounded on external knowledge*, and *evolving with the policy model*. This section introduces the preliminaries of RL with rubrics (§3.1), details our rubric generation and management approach (§3.2), and presents analyses demonstrating its effectiveness (§3.3).

3.1 Preliminaries: RL With Rubric Rewards

Given a question x and its set of corresponding rubrics $\mathcal{R}_x = \{(r_{x,k}, w_{x,k})\}_{k=1}^{K}$, where each $r_{x,k}$ is a rubric (item) and $w_{x,k}$ is its corresponding weight, we assess the quality of the response y with the rubric-based scoring function

$$S(x, y) = \frac{\sum_{k=1}^{K} w_{x,k} \cdot \textsc{Judge}(r_{x,k}, y)}{\sum_{k:\, w_{x,k}>0} w_{x,k}}, \tag{1}$$

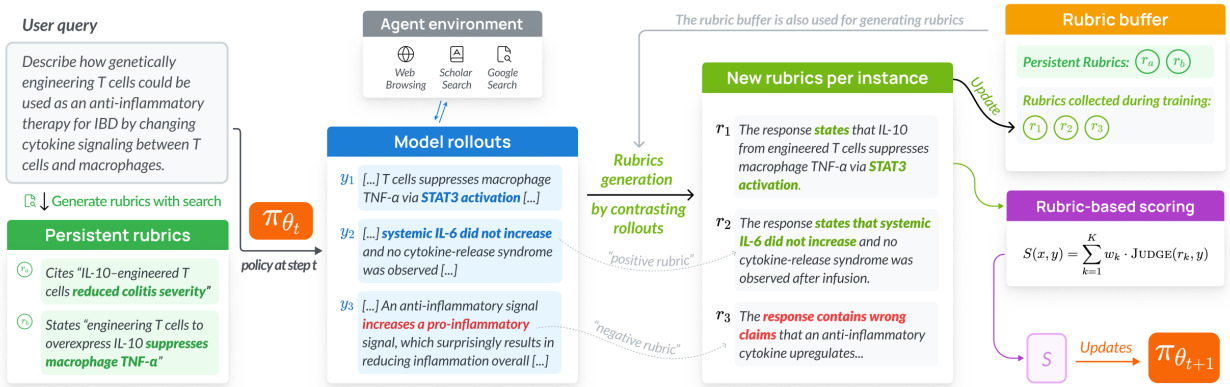

Figure 3 Training with RLER. Given a training instance, the policy LM π_θ generates multiple rollouts via interacting with the environment. We then invoke another LM to create new rubrics based on those rollouts and the current rubrics in the rubric buffer. We score each generation against those rubrics and use that score to update model weights. We then add the new rubrics to the rubric buffer and filter it to retain only a fixed number of rubrics with the highest variance among rollouts.

where, for each rubric $r_{x,k}$, we use a separate judge LM that returns 0, 0.5, or 1 depending on the extent to which $r_{x,k}$ is satisfied by the final answer in y. We compute this rubric score using only the final answer, i.e., regardless of the reasoning and search traces. Note that rubrics are instance-specific, i.e., each question has its own set of rubrics. Rubric weights can be negative to penalize undesirable aspects.

During training, our goal is to optimize the expected reward over questions in the training set. In practice, we optimize this objective using the GRPO (Shao et al., 2024b) algorithm; we discuss further details in Section 4.2.

Existing work instantiates the rubric set $\mathcal{R}_x$ in two main ways. The first approach is to use *general rubrics*, where an LM is prompted to score the response using a single general rubric shared across all instances (Liu et al., 2023; Li et al., 2024, 2025a). However, several works have shown that this approach suffers from reward hacking, where the model exploits biases in the judge rather than learning meaningful behaviors (Gunjal et al., 2025; Zeng et al., 2024). The second approach is to use an LM to generate question-specific rubrics, and then a (potentially separate) LM to perform checklist-style evaluations based on those rubrics (Gunjal et al., 2025). We refer to these rubrics as *closed-book rubrics* since they are generated by a closed-book LM; these are therefore constrained by the generating model's parametric knowledge and might not cover the necessary knowledge to assess DR outputs. In both cases, the rubrics are static: they do not adapt as the policy explores new evidence or behaviors.

3.2 Evolving Rubrics

Designing rubrics for long-form deep-research tasks is particularly challenging. First, long-form questions are often under-specified and admit many plausible ways a response could be good or bad, so a small set of fixed criteria cannot capture all relevant dimensions of quality. Second, DR tasks are highly knowledge-intensive: reliable evaluation requires checking claims against a broad, evolving corpus of world knowledge, rather than relying solely on an LM's parametric knowledge. As a result, closed-book rubrics generated directly by an LM can miss critical evidence, fail to distinguish subtle errors, and are vulnerable to reward hacking by models that exploit judge biases.

We address these challenges by constructing rubrics that *co-evolve* with the policy model and grounds on searched knowledge from the internet. Specifically, instead of infeasibly trying to exhaust all possible desiderata that might not be reachable by the policy model, our method generates rubrics tailored to the current policy model's behaviors, offering on-policy feedback the model can effectively learn from. Furthermore, the rubrics are generated with retrieval, ensuring it can cover the needed knowledge to assess the generation. A detailed illustration on the core RLER training process is in Figure 3.

We next describe the components of our evolving-rubric framework: first, how we initialize rubrics, then how

Algorithm 1 Reinforcement Learning with Evolving Rubrics (RLER)

Require: Dataset $\mathcal{D}$, policy π_θ, rollout size G, max active rubrics $K_{\max}$, rubric generator $\mathcal{G}_{\text{rubric}}$

for each prompt $x \in \mathcal{D}$ **do**
  Generate $\mathcal{R}_x^{\text{persist}} \leftarrow \mathcal{G}_{\text{rubric}}\big(x, \textsc{Search}(x)\big)$ ▷ Generate initial search-based rubrics
  $\mathcal{R}_x^{\text{active}} \leftarrow \emptyset$
for each training step $t = 1, \ldots, T$ **do**
  $\mathcal{R}_x \leftarrow \mathcal{R}_x^{\text{persist}} \cup \mathcal{R}_x^{\text{active}}$
  Rollout with search $\{y_i\}_{i=1}^{G} \sim \pi_\theta(\cdot|x)$
  Generate $\mathcal{R}_x^{\text{new}} \leftarrow \mathcal{G}_{\text{rubric}}(x, \{y_i\}_{i=1}^{G}, \mathcal{R}_x)$; ▷ Generate evolving rubrics by contrasting rollouts
  $\mathcal{R}_x^{\text{active}} \leftarrow \mathcal{R}_x^{\text{new}} \cup \mathcal{R}_x^{\text{active}}$;
  Compute rewards with $\mathcal{R}_x^{\text{persist}} \cup \mathcal{R}_x^{\text{active}}$ and update π_θ (GRPO)
  Compute std of the rewards per rubric
  For $\mathcal{R}_x^{\text{active}}$, remove rubrics with 0 std; keep top-$K_{\max}$ with highest std ▷ Manage rubric buffer

we adapt them online during training, manage the rubric buffer, and finally incorporate auxiliary format and citation rewards, following the pseudocode provided in Algorithm 1.

Initial search-based rubrics. For each training prompt x, we build a customized rubric buffer to store evolving rubrics that are dynamically updated during training. Before training, we initialize the rubric buffer with search-based rubrics. Specifically, for each question x, we first perform a search operation $\textsc{Search}(x)$ to retrieve relevant context from the internet using the original question. We then concatenate the retrieved documents with the question x and feed them into an LM, $\mathcal{G}_{\text{rubric}}$, to produce a set of initial rubrics that will be persistently used throughout RL training: $\mathcal{R}_x^{\text{persist}} = \{R_1, R_2, \ldots, R_{K_s}\}$, where K_s denotes the number of persistent rubrics.

Evolving rubrics during training. During training, we add a new set of evolving rubrics to the active rubric buffer, $\mathcal{R}_x^{\text{active}}$, which are used for scoring. In each training step, for every prompt x and its corresponding set of responses $\{y_i\}_{i=1}^{G}$, where G denotes the number of rollouts, we concatenate the prompt x, all sampled responses $\{y_i\}_{i=1}^{G}$ (including the search context and final answers), and the existing rubric pool $\mathcal{R}_x = \mathcal{R}_x^{\text{persist}} \cup \mathcal{R}_x^{\text{active}}$ as input to $\mathcal{G}_{\text{rubric}}$, obtaining a set of *evolving rubrics* $\mathcal{R}_x^{\text{new}} = \mathcal{G}_{\text{rubric}}\big(x, \{y_i\}_{i=1}^{G}, \mathcal{R}_x\big)$. Specifically, we instruct the LM to generate two types of evolving rubrics: (1) *positive rubrics*, which capture strengths or new, relevant knowledge explored by the current policy but not yet reflected in $\mathcal{R}_x$, and (2) *negative rubrics*, which summarize common undesirable behaviors, such as reward hacking observed across responses—for example, copying retrieved results verbatim to maximize citation precision, or writing an overly long paragraph to gain rubric points and then appending many irrelevant statements with citations to separately boost citation rewards. In both cases, negative rubrics can capture and suppress such behaviors in time. The detailed prompt for evolving rubric generation is provided in Appendix A.1.

Rubric buffer management. Without management, the number of rubrics would grow linearly during training as new evolving rubrics are continuously generated. To maintain a compact yet informative set, we developed a rubric buffer management strategy that filters, merges, and ranks rubrics based on their *discriminative* power. After every GRPO rollout, we score all responses $\{y_i\}_{i=1}^{G}$ using the current active rubrics and obtain rubric-level scores. Rubrics with zero variance in their corresponding rewards are removed as they offer no discriminative value. We then compute the standard deviation for each remaining rubric and rank them by the standard deviation in descending order. To limit evaluation cost, we retain only the top $K_{\max}$ rubrics with the highest standard deviation values.

Format and citation rewards. In addition to evolving rubrics, we introduce two auxiliary rewards—format rewards and citation rewards—to explicitly encourage the model to follow the correct tool usage, reasoning process, and answer format, as well as to provide high-quality citations that support all relevant claims. We detail these two auxiliary rewards in Appendix A.3. We combine the auxiliary rewards with the rubric rewards to form the final training reward, assigning small weights to the auxiliary components.

Rubric Type	Uses Search	Assertive Claims	
		Frac.	Factuality
General Rubrics	✗	0	/
Closed-book Rubrics	✗	0.22	0.94
Initial Rubrics	✓	0.56	0.97
Evolving Rubrics	✓	0.52	1.00

Table 1 The fraction of assertive and factual rubrics. Both the initial search-based rubrics as well as the evolving rubrics (which continue to use search, as they are generated based on the full rollouts including search traces) have a higher proportion of assertive claims compared to closed-book or general rubrics.

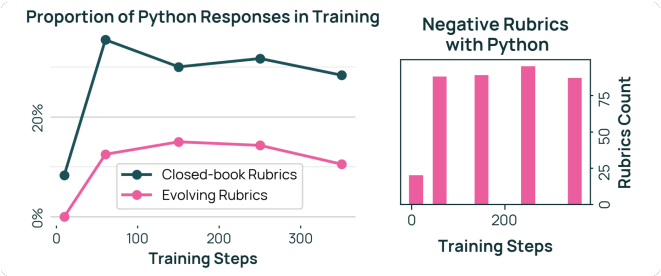

Figure 4 Effect of negative evolving rubrics. Over training, negative evolving rubrics emerge that penalize undesirable behavior such as responding in Python (right), resulting in a reduction in undesirable behaviors over the course of training compared to using a static closed-book rubric that does not specify such undesirable behavior (left).

3.3 How Do Evolving Rubrics Work?

In this section, we show that our initial search-based and evolving rubrics demonstrate desirable properties, such as being specific and adaptive, enabling the verification criteria to more closely approximate the performance of an ideal rubric set compared to naive rubric generation methods.

Search-based and evolving rubrics make verification criteria more concrete and factual. Table 1 compares the specificity of four rubric types. We define a rubric as assertive if it is specific and concrete about what the response should contain (e.g., "The response should mention benchmarks A and B"), and descriptive otherwise (e.g., "The response should discuss benchmarks."). Descriptive rubrics are easier to generate since they do not require factual knowledge, but they often fail to assess response quality accurately, as a model may score well by superficially mentioning a point or even hallucinating facts. We measure the fraction of assertive rubrics and factuality using an LM, with experimental details provided in Appendix B. As shown in Table 1, general rubrics lack specific evaluation criteria, and instance-wise rubrics generated by a closed-book LM are relatively vague (only 22% are assertive). In contrast, initial search-based rubrics and evolving search-based rubrics are more concrete, with over 50% of claims being assertive. These advantages come from search-based rubrics being grounded in retrieved information, and from evolving rubrics being generated using search context, which makes them better suited for training.

Evolving rubrics adjust the evaluation criteria as the policy model evolves. Static rubrics can fail to capture unexpected behaviors or insights emerging during training. As an illustration, we conducted RL training on a single question—"Write a survey paper about RAG." (details in Appendix B). Unexpectedly, some rollouts contained Python code (e.g., Figure 15 in Appendix B), an artifact of the Qwen model that was also previously reported by Shao et al. (2025); this is undesirable but hard for an initial rubric to anticipate. In contrast, evolving rubrics identify these issues and provide negative feedback about irrelevant code, leading to fewer code-containing responses during training (Figure 4).

4 DR Tulu: Training Open Deep Research Agents with RLER

Building on RLER, we introduce **DR Tulu-8B**, a fully open-source deep research agent. In §4.1, we describe how we perform supervised fine-tuning (SFT) on teacher-generated trajectories to resolve the cold-start problem and teach the model basic planning, tool use, and citation skills. §4.2 further refines DR Tulu with online GRPO and RLER using asynchronous tool calls to improve its long-form reasoning and search strategies. Finally, §4.3, we detail our `dr-agent-lib` infrastructure, which is designed to support diverse tools within complex DR workflows and to robustly support high-throughput tool calling in RL training.

4.1 Supervised Fine-Tuning for Cold Start

RLER relies on meaningful exploration over tool-augmented trajectories, but a generic base model does not yet know how to plan, call tools, or cite effectively, so it produces low-quality rollouts. To make this feasible, we first conduct SFT on trajectories produced by a strong teacher model acting as a tool-augmented deep research agent, which gives DR Tulu a reasonable initial search and citation strategy before online RL. Existing open deep research training data are often built for short-form, constrained tasks, so we construct new large-scale SFT trajectories for open-ended queries. We leverage diverse open-source user and task prompts, generate trajectories in an end-to-end manner, and apply lightweight rejection sampling to filter them.

Prompt curation. For long-form, naturally occurring information-seeking questions, we derive prompts from publicly available user–assistant interaction data: **SearchArena** (Miroyan et al., 2025), which contains 24K real-world conversations between users and search-augmented LMs across diverse domains, and **OpenScholar** (Asai et al., 2024), which provides 55K research-oriented queries collected from a deep-research assistant demo. Because real-world queries vary markedly in quality (Cao et al., 2025), we apply a prompt-filtering stage in which an LM rates each prompt on a 1–5 scale (higher is better). For SFT, we mix in a moderate amount of short-form, verifiable QA so the model learns to adapt its style rather than overfit to a single task: we sample from HotpotQA (Yang et al., 2018), TaskCraft (Shi et al., 2025), WebWalker-Silver (Wu et al., 2025a), and MegaScience (Fan et al., 2025), and we generate additional challenging synthetic prompts inspired by PopQA (Mallen et al., 2023). Further curation details appear in the Appendix §C.1.

Trajectory generation and rejection sampling. Given each curated prompt, we generate a full trajectory (model "thinking" traces, tool calls, tool outputs, and the final response) in an end-to-end manner. We provide GPT-5 with a detailed system prompt that defines the deep research workflow and exposes a general web search tool, a paper search tool, and a web browsing tool, and ask it to produce the entire trajectory. Because GPT-5 does not expose its native internal reasoning, we instruct it to generate explicit mock thinking tokens before each tool call or answer tokens. We then apply two lightweight rejection-sampling filters to ensure that the trajectories strictly satisfy our requirements: (1) for all prompts, we verify that trajectories follow the expected tool-calling and answer formats specified in Section 2; and (2) for short-form prompts, we discard trajectories whose final answer does not match the gold answer, following prior work (Jin et al., 2025; Li et al., 2025a). As a result, we curated 16K SFT data points including both short and long-form tasks. Appendix Table 8 shows statistics of our SFT data.

4.2 Online RL with Asynchronous Tool Calls

We then further train DR Tulu using RLER to allow it to explore and improve both its tool-use and answer capabilities, aided by the evolving rubrics described above. We use a customized variant of GRPO (Shao et al., 2024b) with RLER, in which we iteratively generate agentic rollouts using real tool calls, and then score the model's final answer against the evolving rubrics. This allows the model to effectively explore different search strategies in an environment with web-enabled tools, and learn how to further improve its final answers through feedback provided by the evolving rubrics.

Prompt curation. We focus exclusively on long-form questions for RL training. Following the same LM-based filtering procedure used in long-form SFT data curation, we collect approximately 5K new prompts from SearchArena (Miroyan et al., 2025) and OpenScholar (Asai et al., 2024). Additionally, we sample 4K prompts from RaR (Gunjal et al., 2025) to enhance further data diversity.[1] We note that, although we attempted to collect prompts from diverse sources, the questions we gathered are often still out-of-distribution (OOD) relative to those in downstream benchmarks.

RL training. We train our models with GRPO (Shao et al., 2024a), building on the Open-Instruct implementation (Lambert et al., 2025). We use the basic GRPO loss, albeit using *token-level loss aggregation* like DAPO (Yu et al., 2025). We apply two further optimizations: we use sample packing to pack multiple rollouts

[1]For RaR prompts, we use the original rubrics provided by the dataset as the initial set of persistent rubrics, instead of generating new search-based rubrics before training. We still generate evolving rubrics for these prompts.

into single training passes with minimal padding, and use 1-step asynchronous training (Noukhovitch et al., 2024), which means we perform generation and training steps at the same time (training on rollouts from a policy one step behind our current policy), reducing training time. We additionally mask out tool output tokens from the loss, following prior work (Jin et al., 2025). We find using a small KL penalty (0.001) useful for stabilizing training. We provide further hyperparameters in Appendix D.2. After generating rollouts and computing rewards, we perform the rubric buffer management steps described in Section 3.2 before sending the completed samples and rewards to the trainer. We also turned off the citation reward after 650 training steps as we found it converged and did not further add to performance whilst dramatically slowing down RL training (due to the large number of API calls required).

Asynchronous tool calling. When performing tool calls during RL training, we use an asynchronous tool call setup similar to Jiang et al. (2025), wherein tool requests are sent the second a given rollout triggers them, as opposed to waiting for the full batch to finish generating before sending tool calls. Once a tool call is sent, we place that given generation request to sleep, allowing the inference engine to potentially continue to work on generating other responses while waiting for the tool response. This results in generation and tool calling being overlapped wherever possible. Our tool calls are mediated by `dr-agent-lib`, our custom agent infrastructure detailed in the next subsection, which allows tightly controlling the number of concurrent calls made to given APIs, avoiding rate limits and allowing us to cache repeated queries.

4.3 DR Tulu Agent Infrastructure: `dr-agent-lib`

Developing DR agents poses additional infrastructure challenges as the LM needs to call different APIs to search and browse external documents in the generation process. Because DR is sensitive to the quality of the search and browsing tools, it is important to have extensible infrastructure that allows researchers to easily customize and integrate different tools/APIs. The tool backend should also be scalable and robust to handle high-volume concurrent tool calls during RL training. Ideally, it should also be easy to use and support fast iteration of prompts and tool setups during the SFT stage.

To address these challenges, we developed an agent library called `dr-agent-lib` with three key design features. We implement a unified MCP-based tool backend with a collection of local and API-based web search and browsing tools based on the Model Context Protocol (MCP) (see full list in Table 11). The backend is optimized for high concurrency, including a global cache for repeated tool calls and an asynchronous process lock to manage concurrent requests while respecting rate limits. It also features a lightweight and flexible prompt layer, supporting the composition of different search workflows with fine-grained control over prompts and tool-calling configurations. For training, we implement an auto-search workflow (detailed in Appendix E.1) with the following tools: `google_search` (query → top web snippets), `web_browse` (URL → crawled page text), and `paper_search` (query → top relevant paragraphs from open-access papers).

5 Experimental Results

5.1 Experimental Settings

Evaluations. We evaluated deep research agents on four long-form, open-ended benchmarks spanning general-domain, scientific, and medical applications: **HealthBench** (Arora et al., 2025), **ResearchQA** (Yifei et al., 2025), **AstaBench-ScholarQA-CSv2** (SQAv2; Asai et al. 2024; Bragg et al. 2025), and **Deep Research Bench** (DRB; Du et al. 2025). HealthBench targets healthcare deep research; SQAv2 and ResearchQA assess synthesis over up-to-date scientific literature; DRB covers general-domain deep research questions; HealthBench consists of questions posed by individual users for healthcare professionals. For all datasets, responses are expected to be in long form and are evaluated using human-written or human-verified rubric items, following the official evaluation protocols. We also conduct evaluations of DR Tulu on short-form QA in Analysis. For SQAv2 and DRB, we also evaluate fine-grained aspects such as citation precision and recall (the former checks whether statements with citations are actually supported by the cited sources, and the latter checks whether all citation-worthy statements are backed by valid citations), the relevance of answers to the questions, and other properties such as instruction-following. We report both aggregated overall scores and per-aspect scores.

Baselines. We compare against several categories of deep research systems, mirroring the groupings in Table 2. (1) **Open deep-research models**. We compared against four popular open deep research baselines, including ASearcher (7B) (Gao et al., 2025), WebThinker (32B) (Li et al., 2025a), Search-R1 (7B) (Jin et al., 2025), and the concurrent work WebExplorer (8B) (Liu et al., 2025). Notably, none of these methods were evaluated on realistic long-form benchmarks, as their training solely focuses on short-form QA tasks.[2] For long-form tasks, we provide the official evaluation prompts to each model and require a full report-style response. (2) **Fixed-pipeline deep research.** We further evaluated fixed pipeline deep research systems, including WebThinker (report mode) and Ai2 ScholarQA (Singh et al., 2025). Those baselines combine LMs with inference-time only pipelines to generate long-form reports. We ran their official codes with default or recommended configurations. (3) **Closed deep research**: we evaluated OpenAI Deep Research, Perplexity Sonar (reasoning), Perplexity Deep Research, and Claude-Sonnet Search.[3] Lastly, as baselines, we also evaluated Qwen3-8B and QwQ-32B with naive RAG inference and separately with our inference pipeline using our `dr-agent-lib`. We discuss the system prompts and evaluation details in Appendix §E.2.

Training details. We initialize our checkpoint with Qwen3-8b (Yang et al., 2025). For SFT, we run training on one H100 node (8 GPUs) and trained for 5 epochs; the hyperparameters used for SFT are described in Appendix D.1. Overall, finetuning our final SFT model took 136 GPU hours. For RL training, we use the hyperparameters described in Appendix D.2. All training runs were run on 2 H100 nodes (16 GPUs) unless otherwise stated. For rubric scoring, generation, and for citation scoring, we use GPT-4.1-mini (`gpt-4.1-mini-2025-04-14`) as an LM judge. For our final training run, we ran for 15 days, using roughly 6000 GPU hours to take 1000 training steps, or 3.6 epochs over our training data. We found increasing compute did not improve RL training speed, due to being limited by API rate-limits during rollouts. We show the full RL training curves for our final training run in Appendix F.1. We used Crawl4AI,[4] a free open-source tool, for browsing during training time to save costs.

Inference details. We use a single inference pipeline equipped with three tools by default: `google_search`, `web_browse`, and `paper_search`. Following prior work, we use the Serper Search API[5] for `google_search` (Li et al., 2025a) and Jina browsing[6] (Gao et al., 2025; Liu et al., 2025) for `web_browse`, instead of the cheaper Crawl4AI browsing that we used for training. We use Semantic Scholar full-text search API[7] for `paper_search`, which returns paragraphs from relevant papers. We cap the number of tool calls at 10 per rollout to balance efficiency and performance, and we retrieve the top 10 snippets for both `google_search` and `paper_search`. For `web_browse`, whose outputs can be very long, we use Qwen3-8B to summarize the browsed content.

5.2 Main Results

We report overall results across four long-form datasets in Table 2. In addition, Table 3 provides a fine-grained breakdown on the two long-form datasets, SQAv2 and DeepResearchBench.

DR Tulu-8B outperforms all open deep research models on long-form tasks. Across all four open-ended, long-form evaluation benchmarks, DR Tulu-8B consistently outperforms existing open deep research models by more than 10 points on average. In particular, open models designed and trained for constrained, short-form tasks, such as Search-R1 or ASearcher, perform poorly on realistic long-form, report-length generation tasks, yielding low overall scores. The WebThinker models, which are larger (32B), still underperform on long-form generation when used with their default inference pipeline. While the concurrent WebExplorer-8B model is the strongest among prior open deep research systems, both the SFT and RL versions of DR Tulu-8B outperform it across all long-form datasets. Notably, no baselines provide citations, leading to low scores (around 40) on SQAv2, where citation quality is a core evaluation criterion.

[2] WebThinker includes a synthetic report-generation dataset (GLAVE), but evaluation relies solely on LLM-judge ten-scale evaluations (e.g., "comprehensiveness") without explicit citation or evidence checks.

[3] We only consider proprietary systems that provide an API for their deep research systems at the time of writing, which excludes Gemini Deep Research and Grok Deep Research.

[4] https://github.com/unclecode/crawl4ai

[5] https://serper.dev/

[6] https://jina.ai/

[7] https://api.semanticscholar.org/api-docs

	SQAv2	HealthBench	ResearchQA	DRB	Average
Closed Deep Research					
Claude-Sonnet Search	–	–	64.3*	34.5*	–
Perplexity-Sonar (High)	–	–	69.1*	40.7*	–
Perplexity Deep Research	67.3	–	75.3*	42.3*	–
Gemini Deep Research	–	–	68.5*	48.8*	–
GPT-5 + Our Search	61.1	31.1	62.8	50.3	51.3
GPT-5 + Search	74.8	59.5†	78.2†	50.7	65.8
OpenAI Deep Research	79.6	–	72.7*	46.9*	–
Naive RAG					
Qwen3-8B	40.4	16.5	56.1	33.3	36.5
QwQ-32B	41.9	24.5	60.9	40.3	41.9
Open Deep Research Models					
Search-R1-7B	22.2	-0.1	27.9	9.5	14.9
ASearcher-Web-7B	26.9	-13.0	19.4	7.8	10.3
WebExplorer-8B	42.5	33.7	64.8	36.7	44.4
WebThinker-32B-DPO	32.9	11.1	48.6	23.3	28.9
Fixed Pipeline Deep Research					
WebThinker QwQ-32B (report)	45.2	36.5	72.8	37.9	48.1
WebThinker-32B-DPO (report)	46.7	39.4	**74.2**	40.6	50.2
Ai2 ScholarQA - Claude Sonnet	87.7	32.0†	75.0†	36.1	57.7
Open Deep Research (Ours)					
Qwen3-8B + Our Search	57.2	5.9	46.3	18.2	31.9
DR Tulu-8B (SFT)	72.3	38.1	68.5	39.0	53.9
DR Tulu-8B (RL)	**86.7**	**43.7**	71.1	**41.8**	**60.7**

Table 2 Overall results. DR Tulu outperforms all open deep research models, and is competitive with proprietary systems. Rows with a gray background indicate models that use closed models as backbone LMs. Bold indicates the best performance among open models. * denotes scores reported by the original benchmark authors. Except for GPT5 + our tool, we reuse the existing leaderboard results rather than rerunning the evaluations, which would cost a few hundred USD per task; we leave entries as "–" when the original benchmarks do not report the corresponding metric. † denotes that the evaluation was run on a 100-sample subset because the method is expensive. For open models, indicates that the training code is open-sourced, and indicates that the training data is open-sourced. None of the existing open deep research models output citations, so their citation scores on SQAv2 are 0. HealthBench scores can be negative, as HealthBench includes negative rubrics that indicate harmful responses.

DR Tulu-8B outperforms open fixed pipeline deep research heavily engineered for report generation tasks. Open deep research systems with fixed pipelines rely on heavily human-engineered inference workflows to generate long-form reports. In contrast, DR Tulu-8B does not use a separate report-generation pipeline and autonomously decides the length of answers based on given task prompts. WebThinker (report mode) performs competitively on long-form benchmarks, largely due to its pipeline that iteratively drafts and edits answers after running its default search flow with the base model. As shown in Table 4, this dedicated report generation pipeline dramatically increases response length from 90 tokens/answer to over 4000 tokens/answer on average. However, despite producing substantially longer responses and being 4× larger in terms of model size, WebThinker-32B still lags behind DR Tulu-8B on all tasks except ResearchQA. Ai2 ScholarQA, which is specifically designed for scientific literature synthesis and uses Claude Sonnet as the backbone LM, performs competitively with DR Tulu-8B on SQAv2 and surpasses it on ResearchQA. However, this fixed pipeline struggles on DeepResearchBench and HealthBench, as it is tailored to literature synthesis rather than general deep research. As such, despite using a much smaller underlying LM and a single inference pipeline, DR Tulu still achieves the best overall average performance.

We also found that those fixed deep research systems lack the ability to perform simple short-form QA tasks, as even for simple factoid questions (e.g., those in SimpleQA), they attempt to generate a long-form report, generalizing poorly to these types of queries (see appendix E.2 for an example). In contrast, DR Tulu is able to effectively answer such short-form questions, which we more comprehensively evaluate in Section 6.1.

	AstaBench-ScholarQA-CSv2 (SQAv2)				Deep Research Bench (DRB)				
	Rubric	Answer	Cite-P	Cite-R	Comp	Depth	Instruction	Readability	Citation
Closed Deep Research									
Claude-Sonnet Search	-	-	-	-	39.0	37.7	45.8	41.5	93.6
Perplexity Sonar	-	-	-	-	37.4	36.1	45.7	44.7	39.4
Perplexity DR	91.6	92.7	47.3	37.6	40.7	39.3	46.4	44.3	90.2
Gemini Deep Research	-	-	-	-	48.5	48.5	49.2	49.4	81.4
GPT-5+OAI Search	92.3	93.8	67.8	45.6	49.7	51.5	51.6	48.5	63.0
GPT-5+Our Search	74.9	93.2	42.5	33.7	26.7	21.3	41.0	29.4	90.0
OpenAI DR	91.5	95.6	77.4	43.1	46.8	45.2	49.2	47.1	77.9
Naive RAG									
Qwen3-8B	69.2	92.3	-	-	29.4	27.0	40.2	41.1	-
QwQ-32B	77.5	90.3	-	-	38.1	34.8	47.0	44.6	-
Open Deep Research									
Search-R1-7B	9.7	79.0	-	-	5.2	2.1	18.6	16.8	-
ASearcher-7B	13.7	94.0	-	-	5.1	1.7	15.2	11.8	-
WebExplorer-8B	78.6	91.4	-	-	33.7	28.5	45.7	42.2	-
WebThinker-32B-DPO	36.7	94.9	-	-	19.7	12.3	36.8	26.3	-
Qwen3-8B + Our Search	42.8	92.1	53.7	40.3	14.3	8.7	29.5	24.4	**64.7**
DR Tulu-8B (SFT)	81.4	91.0	65.3	51.6	36.3	35.3	45.5	39.5	61.1
DR Tulu-8B (RL)	**84.8**	**95.4**	**90.6**	**76.1**	**39.5**	**39.4**	**47.3**	**41.6**	63.1

Table 3 Performance breakdown for Asta-ScholarQACSv2 and Deep Research Bench. Open deep research models and naive RAG baselines do not provide citations, indicated as "-" in citation columns. Note that Deep Research Bench computes overall score independently from citations while SQAv2 includes citations to compute overall scores.

	Answer	Citations	Tool Calls
GPT-5+Our Search	1487.3	14.7	8.1
GPT-5+OAI Search	2358.7	-	28.1
OpenAI Deep Research	6445.1	79.6	-
Ai2 ScholarQA - Claude Sonnet	2090.5	61.2	1.0
WebExplorer-8B	1250.4	-	9.1
WebThinker-32B	92.2	-	6.9
WebThinker-32B (report)	4416.7	-	8.2
DR Tulu-8B (SFT)	2071.5	38.5	4.6
DR Tulu-8B (RL)	1889.2	35.8	4.3

Table 4 Comparison of model usage statistics on SQAv2. We report answer lengths, tool usage, and citation counts across systems. "-" denotes this information was either not available or it was not applicable.

DR Tulu-8B matches or outperforms proprietary deep research systems. DR Tulu-8B outperforms all closed models on SQAv2. Despite performing similarly to DR Tulu, OpenAI Deep Research generates answers that are roughly three times longer and contain about twice as many citations, as seen in Table 4. DR Tulu additionally outperforms Claude Sonnet Search and Perplexity Sonar (high-reasoning mode) on all benchmarks, and exceeds Perplexity Deep Research on three datasets. These proprietary systems are also significantly more expensive; OpenAI Deep Research costs USD 1.80 per query on SQAv2, and Ai2 ScholarQA (using Claude Sonnet) costs USD 1.30 per query. In comparison, DR Tulu costs approximately USD 0.00008 per query, assuming the user has already paid the hardware and hosting costs (only including the cost of API calls in the query cost).[8] Even if the model calls the most expensive API (Serper search) the maximum number of times allowed during evaluation (10 times), this gives a maximum cost of USD 0.0075 per query.

RLER improves deep research quality across diverse aspects. We break down the performance of models on two long-form datasets, SQAv2 and DeepResearchBench in Table 3. Comparing DR Tulu (SFT) and DR

[8]We find that DR Tulu calls tools about 4.3 times on average per SQAv2 query. It calls `paper_search` 96.8% of the time, `google_search` 2.4% of the time, and `web_browse` 0.9% of the time. S2 API is free to use, we estimate Serper search as USD 0.00075 per query and Jina browse as USD 0.00005 per query. As such, the average query costs USD 0.000079.

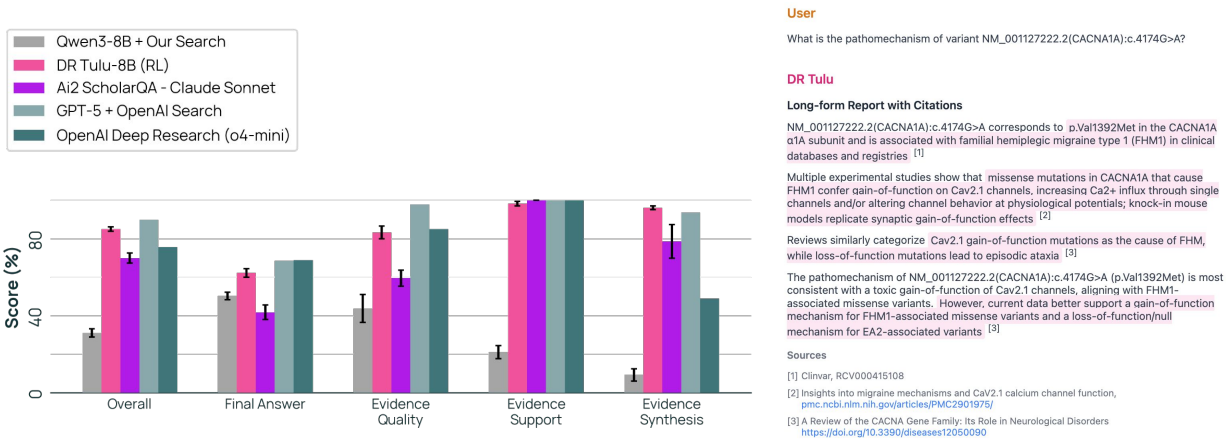

Figure 5 Researching pathogenic gene variants in GeneticDiseasesQA. Left: DR Tulu-8B matches or outperforms closed DR models on GeneticDiseasesQA, a benchmark that we constructed to reflect a real-world medical genetics task that is out-of-domain to our training data. We were unable to compare to other open deep research models and naive RAG baselines as they do not produce evidence citations, which are critical to GeneticDiseasesQA. Results for DR Tulu-8B (RL) and Qwen3-8B + Our Search and Ai2 ScholarQA are reported as the average of 3 trials. Due to the expense and long inference times for OpenAI Deep Research and GPT-5+OpenAI Search, we report results from only 1 trial. Right: Example question and response snippets generated by DR Tulu-8B.

Tulu (RL), we observe consistent gains from RLER across multiple aspects, including rubric coverage (+3.4 points), comprehensiveness (+3.2), and depth of response (+4.1). RLER also yields large improvements in citation precision and recall on SQAv2 (+25.3 and +25.5 points, respectively). These results highlight that RLER can effectively improve deep research responses along both content and attribution dimensions.

5.3 Application: Researching Pathogenic Gene Variants

We next applied DR Tulu to a new out-of-domain deep research task motivated by a real-world challenge in medical genetics: investigating disease-causing gene variants. In clinical settings, accurately identifying and interpreting variants on patient genetic tests is essential for diagnosing genetic disorders and recommending personalized treatment strategies (Cheerie et al., 2025). Although whole genome sequencing technologies have improved the diagnostic yield of rare genetic disease (Nurchis et al., 2023), aggregating sparse information across databases, research papers, and case reports is still a key bottleneck in variant interpretation (Mastrianni et al., 2025). This combination of information synthesis and biological reasoning, together with the need for evidence attribution and interpretability in tools for medical decision making, makes this task a natural fit for automated deep research systems.

GeneticDiseasesQA. We create a new evaluation dataset, **GeneticDiseasesQA**, consisting of 47 questions derived from expert-curated information about 24 disease-causing genetic variants. Question topics were centered around finding information that genetics experts use to assess variant eligibility to various gene therapy strategies (Cheerie et al., 2025), which involve reasoning about the molecular properties, disease-causing mechanisms, therapeutic approaches, etc. We show an example in Figure 5. For each question, we prompt deep research systems to generate a long-form report that both answers the question and provides supporting evidence for each claim. We define the following criteria for evaluation, again using GPT-4.1 as a judge to score each category: *Final Answer* indicates whether the expert-annotated fact(s) were mentioned in the response. *Evidence Quality* indicates if the type of evidence requested in the query (e.g. functional assays in patient-derived cells) is present within the cited statements. *Evidence Synthesis* indicates whether or not there was at least one statement describing the relationship between multiple sources. More details are in Appendix E.4.

Results. We compare DR Tulu-8B (RL) with Qwen3-8B with our search tool, Ai2 ScholarQA, GPT-5 + OpenAI Search, and Open AI Deep Research using o4-mini.[9] We do not include the open deep research models or fixed pipeline deep research models that do not provide citations, as it is a key aspect of the evaluation. Figure 5 reports results on the variant analysis task. DR Tulu-8B consistently outperforms the base Qwen3-8B model across all categories, and furthermore surpasses Ai2 ScholarQA Agent—which runs on top of Claude Sonnet and is specifically designed for scientific literature synthesis—by a large margin on overall score, evidence quality, and the final answer. It also outperforms OpenAI Deep Research, especially on evidence synthesis, and is competitive with the state-of-the-art GPT-5 + OpenAI Search system. These results show that DR Tulu-8B generalizes to real-world DR tasks that are unseen and out-of-domain to the training data.

6 Analysis

6.1 Evaluation on Short-form QA Tasks

In this paper, we focus on long-form, open-ended deep research, and our RLER training is applied exclusively to long-form tasks. However, our SFT mixture deliberately includes short-form, verifiable QA tasks that still require search, so the model learns to handle both concise answers and multi-paragraph long answers. We therefore examine how well our SFT and RL models perform on standard short-form QA benchmarks to determine how well our approach generalizes to varied query types.

Datasets. Our short-form evaluation suite comprises **SimpleQA** (Wei et al., 2024), **WebWalkerQA** (Wu et al., 2025a), and **2Wiki** (Ho et al., 2020). For short-form QA, we follow prior work (Li et al., 2025a; Wei et al., 2024) and use an LLM judge to determine answer correctness against the annotated gold, reporting judge-based accuracy. We evaluate performance with Pass@1. We randomly selected 1000 questions each from SimpleQA and 2Wiki for efficient evaluation.

Results. Table 5 reports the short-form QA results. We find that DR Tulu performs competitively on short-form, verifiable QA benchmarks. Our SFT stage greatly improves over the Qwen3-8B + `dr-agent-lib` baseline, demonstrating the effectiveness of our SFT data for short-form QA. Interestingly, although the RL stage uses only long-form prompts and explicitly optimizes long-form generation quality, DR Tulu (RL) achieves further gains across datasets, improving the overall average by 4.4 points. This suggests that our RL recipe is effective not only for long-form deep-research tasks, but also generalizes well to short-form QA.

	SimpleQA	2Wiki	WebWalker	Avg.
Naive RAG				
Qwen3-8B	52.6	18.9	8.8	26.8
QwQ-32B	57.2	34.2	10.1	33.8
Open Deep Research (Ours)				
Qwen3-8B + Our Search	70.5	44.0	27.9	47.5
DR Tulu-8B (SFT)	75.5	66.5	31.9	58.0
DR Tulu-8B (RL)	80.1	68.0	39.1	62.4

Table 5 Short-form results. We report short-form performance for our SFT and RL variants to analyze how each training stage affects short-form behavior. All scores are computed from top-1 predictions and evaluated with GPT-4.1 as the LLM judge under a unified evaluation pipeline.

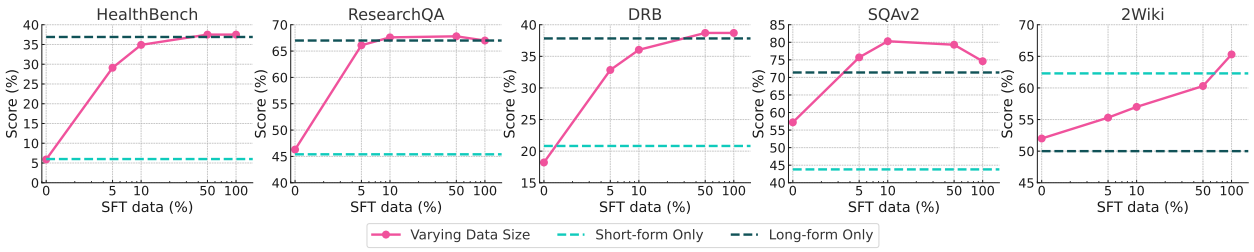

Figure 6 Ablation of SFT training data. We ablate SFT training data in terms of mixture of data and scale of training data. We train models with varying size of SFT data (5%, 10%, 100%; 0% indicates the Qwen3-8B + `dr-agent-lib` results) as well as two SFT subsets, long-form data only (LF only) and short-form data only (SF only).

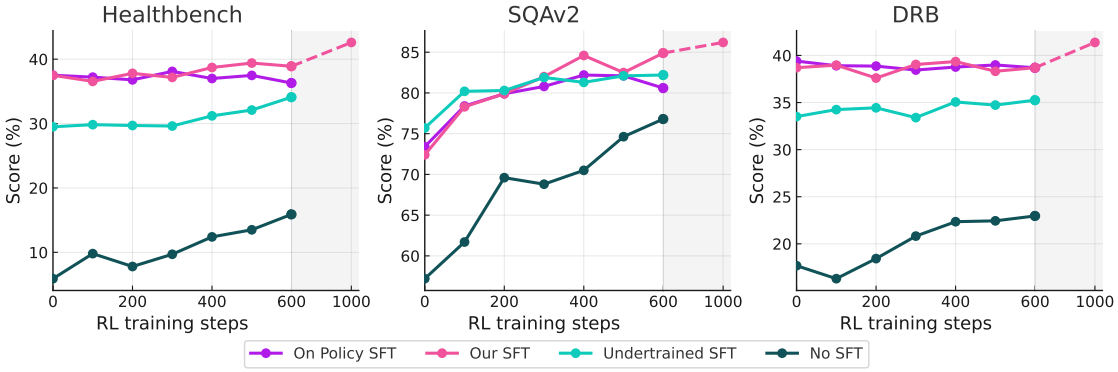

Figure 7 Performance of different starting point models throughout RL training. We vary the SFT model used at the start of RL training, and then train using the same data and hyperparameters. RL gives smaller improvements the stronger the starting point model. Further RL training further improves results.

6.2 Analysis on Training

Including both long-form and short-form SFT data is helpful. Figure 6 reports an ablation of the SFT training mixture. We analyze (i) removing the long-form or short-form subsets entirely, and (ii) subsampling the full corpus uniformly at random. Many existing open deep research models are trained only on short-form QA, assuming that short-form supervision will transfer to long-form synthesis, while other systems are optimized purely for long-form reporting and struggle to produce concise answers. Our ablations show that neither extreme is ideal: relying only on short-form or only on long-form data is suboptimal. Removing long-form data substantially degrades performance on all four long-form datasets. Conversely, removing short-form data leaves HealthBench, ResearchQA, and DRB largely unchanged, but noticeably reduces performance on 2Wiki and SQAv2. This suggests that short-form supervision alone does not automatically yield strong long-form, open-ended deep research performance. Although our primary focus is long-form deep research, we retain a modest short-form component so that the model remains more general-purpose and handles short-form questions well, without sacrificing long-form performance.

SFT scaling: saturated gains for long-form, continued gains for short-form. All tasks show clear gains as we initially increase the amount of SFT data. On 2Wiki, performance continues to improve up to 100% of the data, suggesting that short-form tasks may benefit more directly from scaling SFT data. On long-form tasks, we already observe substantial gains with just 5% of the training data, indicating that even a relatively small amount of high-quality long-form trajectory data is highly beneficial. However, improvements saturate after 50%, highlighting that, unlike short-form QA, further boosting deep research agent performance on open-ended tasks is challenging if we only scale SFT data. While we observe a slight drop in SQAv2 from using the

[9]Note that Evidence Support is computed using supporting snippets. For OpenAI models that do not return snippets, we instead retrieve the full webpage content via Jina browsing using the provided URLs, which differs from how we compute Evidence Support for DR Tulu, Qwen3-8B, and Ai2 ScholarQA Agent. Nevertheless, all systems perform reasonably well on Evidence Support.

	SQAv2	HealthBench	ResearchQA	DRB	Avg.
Qwen3 8B + SFT v0.1	73.4	37.5	69.6	39.4	55.0
+ RL w/ General rubrics (500 steps)	80.6 (+7.2)	36.0 (-1.5)	65.0 (-4.6)	34.1 (-5.3)	53.9 (-1.1)
+ RL w/ Closed-book rubrics (500 steps)	83.2 (+9.8)	34.8 (-2.7)	65.0 (-4.6)	37.6 (-1.8)	55.2 (+0.2)
+ RL w/ Initial search-based rubrics (500 steps)	82.8 (+9.4)	37.9 (+0.4)	66.9 (-2.7)	39.3 (-0.1)	56.7 (+1.7)
Qwen3 8B + SFT v0.2	76.1	34.3	67.8	38.5	54.2
+ RL w/ Initial search-based rubrics (650 steps)	83.2 (+7.1)	39.3 (+5.0)	67.8 (+0.0)	39.0 (+0.5)	57.3 (+3.1)
+ RL w/ Evolving rubrics (650 steps)	84.9 (+8.8)	38.9 (+4.6)	68.7 (+0.9)	40.2 (+1.7)	58.2 (+4.0)

Table 6 Ablation on rubrics. We ablate the rubrics used during RL training, and the effect of adding evolving rubrics. We find that using simple rubrics can actually hurt performance, while search-based rubrics perform best. Further adding evolving rubrics gives a one point improvement on average. For compute reasons, we stopped this ablation after 650 steps and switched to training with evolving rubrics for longer. SFT v0.1 and v0.2 refer to different intermediate SFT mixtures developed during the project, which used similar mixtures to our final SFT checkpoint.

full SFT dataset, we found this was largely driven by citation scores dropping, while ingredient and answer-based metrics remained similar.[10] As such, we used the full-size SFT dataset to maintain strong short-form QA abilities, and as we found RL training was able to significantly recover SQAv2 citation performance.

RL benefits from stronger SFT models and longer training. We ablate the effect of using different SFT cold start datasets on RL in Figure 7, tracing performance up to 600 training steps. Beginning RL directly from Qwen3 (no SFT cold start) dramatically improves scores over Qwen3-8B with no training, but still underperforms using even a small amount of high-quality SFT data (5% of our full mixture) as cold-start data for the RL training. Using a larger amount of SFT data (i.e., our full SFT mixture) further improves performance. We additionally explore augmenting the SFT data with an extra "on-policy" SFT stage. Specifically, we run our trained model on randomly sampled prompts, apply rejection sampling to discard trajectories that do not achieve high scores on search-based rubric verification and citation verification (details in Appendix C.3), and then use the remaining trajectories for further SFT. While this slightly boosts SFT model performance, we find it ultimately weakens performance later on during RL training, underperforming using our regular SFT mixture on Healthbench and SQAv2. We also found that extended RL training was crucial to performance: in some cases, evaluations that initially seemed flat (e.g., DRB) improved with extended RL training up to 1000 steps. We also note that we found that higher train reward (i.e., reward during RL training) did not necessarily match with higher downstream reward, see Appendix F.4 for details.

Search-based rubrics outperform closed-book rubrics. We ablate the effect of using different static rubric setups (i.e., without adding evolving rubrics) during RL training in Table 6. We run RL training for 500 steps on top of an intermediate SFT checkpoint using four different rubric setups: (1) general rubrics, in which we use a simple prompt and LM judge to score model outputs (see Appendix D.3 for prompt); (2) closed-book rubrics, which are generated without access to any search information; (3) search-based rubrics, which are generated with knowledge from an initial search (See Section 3 for details). For these runs, we only used training samples from ScholarQA. We find that using search-based rubrics performs best overall, while using a single general rubric for all samples actually results in lower performance than the SFT starting point.

Evolving rubrics improve over initial rubrics alone. We additionally ablate using evolving rubrics on top of search-based rubrics in Table 6, training for 650 steps (due to compute and budget limitations) over an intermediate SFT checkpoint. We find that using evolving rubrics during training can provide a boost over search rubrics alone in all long-form evaluations except Healthbench. We expect that this gap would further widen with longer training, as evolving rubrics further capture new knowledge explored by the model.

[10] We hypothesize the citation score drop is due to the presence of model calls not supported by our pipeline in the SFT data. These calls take the form of XML tags that the citation scorer picks up and penalizes (e.g. `<Model name=Anthropic version=claude-3-7-sonnet-20250219>`). These comprised a small portion of our data, and as such is not picked up by smaller SFT subsets, but is picked up when training on the full dataset. We will further investigate this drop in future work.

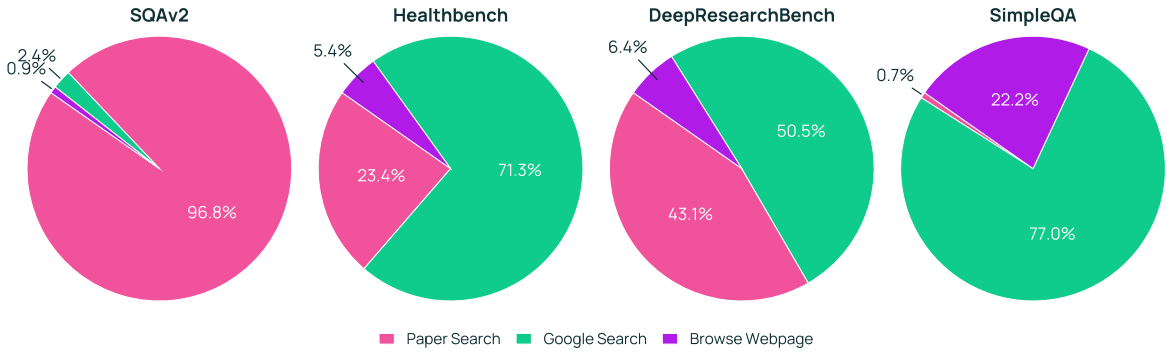

Figure 8 Distribution of tool calls for SQAv2 (science), HealthBench (healthcare), Deep Research Bench (general domain) and SimpleQA (factoid, short-form QA). DR Tulu can adaptively choose effective tools for different tasks, relying more on `paper_search` for scientific questions (SQAv2), and more on `google_search` for general-domain questions (SimpleQA).

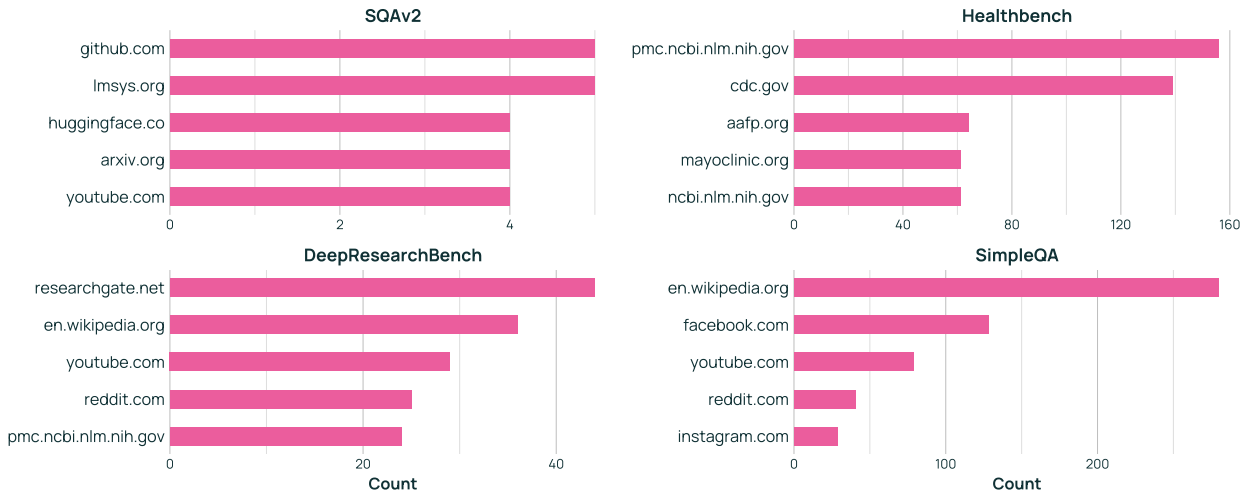

Figure 9 Distribution of domains among web search results for SQAv2 (science), HealthBench (healthcare), Deep Research Bench (general domain) and SimpleQA (factoid, short-form QA). We show top domains returned by the `google_search` tool. Calculations are based on 100 samples from each task. These top domains match the evaluation domain; e.g., when evaluating on Healthbench, DR Tulu searches more for medical domain websites.

6.3 Analysis on Inference

Tool and domain usage adapt to each task's information needs. Figure 8 shows that `snippet_search` (our scientific-paper search) dominates on SQAv2, consistent with its focus on literature understanding. In contrast, Google Search is the primary tool for HealthBench, DeepResearchBench, and SimpleQA, reflecting the broader, open-web information needs of these tasks. `web_browse` is used comparatively less across all datasets, serving as a follow-up tool for reading full pages when snippet context is insufficient.

Figure 9 further confirms task-specific retrieval behavior. HealthBench emphasizes authoritative biomedical and public-health sites (e.g., `cdc.gov`, `pmc.ncbi.nlm.nih.gov`, ncbi.nlm.nih.gov, `mayoclinic.org`). DeepResearchBench mixes technical and policy sources (e.g., `researchgate.net`, `oecd.org`, `github.com`), consistent with deeper, exploratory research tasks. SimpleQA is dominated by general reference and social/information platforms (e.g., `en.wikipedia.org`, `facebook.com`, `youtube.com`). Overall, tool usage and surfaced domains align with each dataset's information demands: literature-centric tasks favor scientific search and scholarly venues, whereas open-domain tasks lean on general web search and broad reference sites.

Model	Size	Long-form	Multi-Search	Citations	Open-Source			
					Train. Code	Eval Code	Train. Data	Model Ckpt
Search-R1	7B	✗	✗	✗	✓	✓	✓	✓
WebThinker	32B	✓*	✗	✗	✗	✓	✗	✓
WebExplorer	8B	✗	✗	✗	✗	✓	✗	✓
ASearcher	7,14,32B	✗	✗	✗	✓	✓	✓	✓
SFR DR	8B	✗	✗	✗	✗	✗	✗	✗
Ai2 ScholarQA	–	✓	✗	✓	–	✓	–	–
WebWeaver	–	✓	✓	✗	–	✓	–	–
SFR EDR	–	✓	✓	✗	–	✓	–	–
DR Tulu	8B	✓	✓	✓	✓	✓	✓	✓

Table 7 Comparison with existing deep research systems. We compare our method with existing open deep research models, namely Search-R1 (Jin et al., 2025), WebThinker (Li et al., 2025a), WebExplorer (Liu et al., 2025), SFR-DeepResearch (SFR-DR) (Nguyen et al., 2025), Ai2 ScholarQA (Singh et al., 2025), SFT-Enterprise Deep Research (SFR-EDR) (Prabhakar et al., 2025) and WebWeaver (Li et al., 2025b). * indicates tested on long-form evaluation benchmarks using a specifically designed long-form report agent workflow. Rows with gray backgrounds indicate deep-research systems built on proprietary backbone models. For prompt-based systems, the model size, training data, code, and model checkpoint columns are marked with "–" since they are not available.

7 Related Work

Deep research agents. Rapid adoption of commercial deep research systems has spurred numerous open efforts, but most target verifiable, short-form generation (e.g., factoid QA or lightweight web-browsing tasks). Inspired by the success of scaling online RL on verifiable domains such as code and math, many methods follow a similar recipe. For example, Search-R1 (Jin et al., 2025) applies GRPO to enhance search capabilities and is trained primarily on short-form question answering. Such approaches have been explored in many recent followup studies, including WebExplorer (Liu et al., 2025). In contrast, WebThinker (Li et al., 2025a) employs DPO and proposes a report-generation workflow. Nevertheless, the majority of these works train and evaluate only on short-form outputs. Furthermore, most open deep-research systems either rely on a single web search tool or train separate models per search backend (Gao et al., 2025). In expert domains (e.g., healthcare, science), we find that combining multiple search tools yields substantial gains. In addition, existing open systems typically omit explicit citations, unlike proprietary counterparts.

Furthermore, many of these models do not fully open-source their training data or training codebases, making it difficult to further analyze or improve their training. A complementary line of work builds deep-research agents by carefully designing fixed long-form generation pipelines, often on top of state-of-the-art proprietary models, including WebWeaver (Li et al., 2025b), SFT-Enterprise Deep Research (SFR-EDR) (Prabhakar et al., 2025), and Ai2 ScholarQA (Singh et al., 2025). Leveraging strong backbone LMs, these systems overcome some of the aforementioned limitations and are evaluated primarily on long-form generation tasks. However, their fixed pipelines sacrifice flexibility in inference flow and output style (e.g., always producing long-form reports even for simple factoid questions), and they still do not provide a clear path toward *open*, end-to-end trainable deep-research models. Table 7 summarizes these gaps. To our knowledge, our system is the first fully open deep-research framework that (i) is trained and rigorously evaluated on realistic long-form tasks, (ii) natively supports multi-tool search rather than single-tool or siloed models, and (iii) produces evidence-linked citations. We fully open-source our training and evaluation code bases, data, and a new infra to support flexible deep research agent developments.

Rubric design for long-form generation tasks. Prior work uses human-written rubrics for evaluation (Arora et al., 2025; Asai et al., 2024), but it is costly and not scalable when applied for training. RaR (Gunjal et al., 2025) proposed to use rubrics as rewards and generate instance-wise rubrics based on reference answers from an advanced model—o3. However, these rubrics are static and usually generated by the same model, which can only slow down reward hacking but does not resolve the issue. In addition, these approaches rely on the capabilities of the model used to generate reference answers, whose knowledge is limited and not up to date, and thus cannot meet the needs of DR tasks. Concurrent work (Rezaei et al., 2025) explores generating online rubrics by contrasting pairwise model rollouts in a closed-book setting. This approach echoes the design

principle of our evolving rubrics but lacks grounding in external knowledge, which leads to exploitation (reshaping model behavior based solely on its internal knowledge) rather than exploration (integrating new external knowledge while also exploiting existing knowledge). Another concurrent work, RLAC (Wu et al., 2025b), explores training a critic to propose a likely incorrect fact that serves a similar role to a rubric for factuality tasks. Compared with concurrent works, our approach focuses on a more challenging setup—DR tasks—and generates rubrics that both co-evolve with the policy model and remain grounded in external knowledge, enabling prolonged RL training with an evolving verifier.

8 Discussion and Future Work

Evolving rubrics adapt the verifier based on the policy model's capabilities. At each training step, we update our rubrics by contrasting the model's current rollouts, which helps the new rubric criteria better distinguish those outputs. We can view this as making the training difficulty adaptive to the model's evolving behavior. This approach aligns with the idea of training in *adaptive environments*, which has been previously explored by adjusting prompts during training (Zeng et al., 2025). In contrast, we adapt the environment by updating the verifier (rubrics). Future work may consider jointly adapting both prompts and rubric criteria to further improve training efficiency.

Interpreting evolving rubrics from a knowledge coverage perspective. Figure 10 shows an abstract visualization of the knowledge coverage of different rubric types. Search expands the rubric knowledge coverage beyond the capacity of the parametric knowledge of the rubric generator (an LM) for generating the initial rubrics. Furthermore, evolving rubrics generated during training fold in new evidence discovered by the deep research policy during training rollouts, capturing knowledge that requires complex reasoning and planning to obtain, and allowing the evaluation criteria to evolve with the model's distribution.

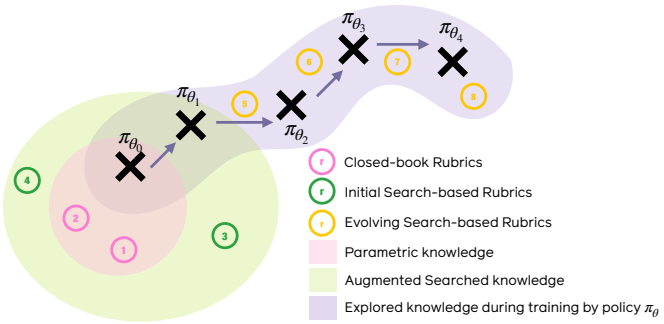

Figure 10 Knowledge coverage relationship visualization. An abstract visualization of the knowledge coverage relationship between closed-book rubrics, initial search-based rubrics, and evolving search-based rubrics.

A new dimension of scaling verifier compute: providing more context to the judge. Another perspective on RLER is that it creates a new way to scale the compute used by the verifier/reward model. While prior work focuses on increasing the reasoning tokens used by the reward model given limited context (Guo et al., 2025; Chen et al., 2025b), we instead enrich the information available to it—the policy's reasoning process, the external knowledge grounding its claims, and possible alternative responses—resulting in a potentially more effective and meaningful use of verifier compute under a fixed budget.

The train-test mismatch challenge. When developing DR Tulu, we found that models that achieved the highest training reward did not necessarily achieve the highest downstream evaluation performance, although within the same run, higher training rewards usually correlated with better downstream performance; see Appendix F.4 for more details. We conjecture that this stems from a mismatch between the tasks, rubrics, and evaluation setups of the external benchmarks vs. what we used for training. For instance, RL training uses a judge that differs from the judges used in downstream evaluations, which can lead to reward hacking toward preferences specific to the training-time judge. Moreover, external benchmarks often use expert-crafted or generated rubrics that may emphasize aspects not captured in our training rubrics. Some of these rubrics might not be evident from the question alone, making it difficult for models like ours that are not trained for those benchmarks in particular. On the other hand, this also highlights the utility of having fully open DR models, like DR Tulu, as they can be straightforwardly customized and adapted for downstream tasks.

Author Contributions

DR Tulu is a team effort. Here, we describe each author's primary contributing roles in the project, with bolded authors taking the lead within each section:

- Project leads: **Rulin Shao**, Akari Asai
- Core contributors: Rulin Shao, Akari Asai, Shannon Shen, Hamish Ivison, Varsha Kishore, Jingming Zhuo
- RLER method development: **Rulin Shao**, Hamish Ivison, Shannon Shen
- DR Tulu data: **Akari Asai**, Rulin Shao, Jingming Zhuo, Varsha Kishore, Shannon Shen, Luca Soldaini
- DR Tulu training: **Hamish Ivison**, **Rulin Shao**, Akari Asai, Shannon Shen
- Infrastructure: **Shannon Shen**, **Hamish Ivison**, Rulin Shao, Luca Soldaini, Tyler Murray, Varsha Kishore
- Evaluations and baselines: Varsha Kishore, Shannon Shen, Rulin Shao, Akari Asai, Jingming Zhuo, Hamish Ivison, Xinran Zhao
- GeneticDiseasesQA benchmark creation and evaluations: **Molly Park**, Samuel Finlayson
- Project mentorship: Hannaneh Hajishirzi, Pang Wei Koh, Yoon Kim, Luke Zettlemoyer, Sherry Tongshuang Wu, Scott Yih, David Sontag, Faeze Brahman, Luca Soldaini, Pradeep Dasigi, Sewon Min.

Core contributors made sustained, significant contributions throughout the project. All authors contributed to project discussions, experiment planning, and writing the paper.

Acknowledgments

This work was supported by the Singapore National Research Foundation and the National AI Group in the Singapore Ministry of Digital Development and Information under the AI Visiting Professorship Programme (award number AIVP-2024-001), the AI2050 program at Schmidt Sciences, and the DARPA SciFy program (Agreement No. HR00112520300). We thank Zhiyuan Zeng, Rui Xin, Stella Li, and Doug Downey for helpful discussions and feedback on the draft.

References

Rahul K Arora, Jason Wei, Rebecca Soskin Hicks, Preston Bowman, Joaquin Quiñonero-Candela, Foivos Tsimpourlas, Michael Sharman, Meghan Shah, Andrea Vallone, Alex Beutel, et al. Healthbench: Evaluating large language models towards improved human health. *arXiv preprint arXiv:2505.08775*, 2025.

Akari Asai, Jacqueline He, Rulin Shao, Weijia Shi, Amanpreet Singh, Joseph Chee Chang, Kyle Lo, Luca Soldaini, Sergey Feldman, Mike D'arcy, et al. Openscholar: Synthesizing scientific literature with retrieval-augmented lms. *arXiv preprint arXiv:2411.14199*, 2024.

Jonathan Bragg, Mike D'Arcy, Nishant Balepur, Dan Bareket, Bhavana Dalvi, Sergey Feldman, Dany Haddad, Jena D Hwang, Peter Jansen, Varsha Kishore, et al. Astabench: Rigorous benchmarking of ai agents with a scientific research suite. *arXiv preprint arXiv:2510.21652*, 2025.

Tianyu Cao, Neel Bhandari, Akhila Yerukola, Akari Asai, and Maarten Sap. Out of style: Rag's fragility to linguistic variation. *arXiv preprint arXiv:2504.08231*, 2025.

D. Cheerie, M. M. Meserve, D. Beijer, C. Kaiwar, L. Newton, A. L. Taylor Tavares, A. S. Verran, E. Sherrill, S. Leonard, S. J. Sanders, E. Blake, N. Elkhateeb, A. Gandhi, N. S. Y. Liang, J. T. Morgan, A. Verwillow, J. Verheijen, A. Giles, S. Williams, M. Chopra, L. Croft, H. S. Dafsari, A. E. Davidson, J. Friedman, A. Gregor, B. Haque, R. Lechner, K. A. Montgomery, M. Ryten, E. Schober, G. Siegel, P. J. Sullivan, E. F. Whittle, B. Zardetto, T. W. Yu, M. Synofzik, A. Aartsma-Rus, G. Costain, M. C. Lauffer, and N=1 Collaborative. Consensus guidelines for assessing eligibility of pathogenic dna variants for antisense oligonucleotide treatments. *American Journal of Human Genetics*, 112(5):975–983, May 2025. doi: 10.1016/j.ajhg.2025.02.017. Epub 2025 Mar 25.

Liang Chen, Xueting Han, Li Shen, Jing Bai, and Kam-Fai Wong. Beyond two-stage training: Cooperative sft and rl for llm reasoning, 2025a. https://arxiv.org/abs/2509.06948.

Xiusi Chen, Gaotang Li, Ziqi Wang, Bowen Jin, Cheng Qian, Yu Wang, Hongru Wang, Yu Zhang, Denghui Zhang, Tong Zhang, et al. Rm-r1: Reward modeling as reasoning. *arXiv preprint arXiv:2505.02387*, 2025b.

Jonathan H Clark, Eunsol Choi, Michael Collins, Dan Garrette, Tom Kwiatkowski, Vitaly Nikolaev, and Jennimaria Palomaki. Tydi qa: A benchmark for information-seeking question answering in ty pologically di verse languages. *Transactions of the Association for Computational Linguistics*, 8:454–470, 2020.

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.

Run-Ze Fan, Zengzhi Wang, and Pengfei Liu. Megascience: Pushing the frontiers of post-training datasets for science reasoning. *arXiv preprint arXiv:2507.16812*, 2025.

Jiaxuan Gao, Wei Fu, Minyang Xie, Shusheng Xu, Chuyi He, Zhiyu Mei, Banghua Zhu, and Yi Wu. Beyond ten turns: Unlocking long-horizon agentic search with large-scale asynchronous rl. *arXiv preprint arXiv:2508.07976*, 2025.

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Yunzhong He, Bing Liu, and Sean Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains. *arXiv preprint arXiv:2507.17746*, 2025.

Jiaxin Guo, Zewen Chi, Li Dong, Qingxiu Dong, Xun Wu, Shaohan Huang, and Furu Wei. Reward reasoning model. *arXiv preprint arXiv:2505.14674*, 2025.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In Donia Scott, Nuria Bel, and Chengqing Zong, editors, *Proceedings of the 28th International Conference on Computational Linguistics*, pages 6609–6625, Barcelona, Spain (Online), December 2020. International Committee on Computational Linguistics. doi: 10.18653/v1/2020.coling-main.580. https://aclanthology.org/2020.coling-main.580/.

Dongfu Jiang, Yi Lu, Zhuofeng Li, Zhiheng Lyu, Ping Nie, Haozhe Wang, Alex Su, Hui Chen, Kai Zou, Chao Du, et al. Verltool: Towards holistic agentic reinforcement learning with tool use. *arXiv preprint arXiv:2509.01055*, 2025.

Bowen Jin, Hansi Zeng, Zhenrui Yue, Jinsung Yoon, Sercan Arik, Dong Wang, Hamed Zamani, and Jiawei Han. Search-r1: Training llms to reason and leverage search engines with reinforcement learning. In *COLM*, 2025.

Kalpesh Krishna, Aurko Roy, and Mohit Iyyer. Hurdles to progress in long-form question answering. 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 4940–4957, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.393. https://aclanthology.org/2021.naacl-main.393/.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James Validad Miranda, Alisa Liu, Nouha Dziri, Xinxi Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Christopher Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tulu 3: Pushing frontiers in open language model post-training. In *Second Conference on Language Modeling*, 2025. https://openreview.net/forum?id=i1uGbfHHpH.

Haitao Li, Qian Dong, Junjie Chen, Huixue Su, Yujia Zhou, Qingyao Ai, Ziyi Ye, and Yiqun Liu. Llms-as-judges: a comprehensive survey on llm-based evaluation methods. *arXiv preprint arXiv:2412.05579*, 2024.

Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yutao Zhu, Yongkang Wu, Ji-Rong Wen, and Zhicheng Dou. Webthinker: Empowering large reasoning models with deep research capability. *arXiv preprint arXiv:2504.21776*, 2025a.

Zijian Li, Xin Guan, Bo Zhang, Shen Huang, Houquan Zhou, Shaopeng Lai, Ming Yan, Yong Jiang, Pengjun Xie, Fei Huang, et al. Webweaver: Structuring web-scale evidence with dynamic outlines for open-ended deep research. *arXiv preprint arXiv:2509.13312*, 2025b.

Junteng Liu, Yunji Li, Chi Zhang, Jingyang Li, Aili Chen, Ke Ji, Weiyu Cheng, Zijia Wu, Chengyu Du, Qidi Xu, et al. Webexplorer: Explore and evolve for training long-horizon web agents. *arXiv preprint arXiv:2509.06501*, 2025.

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. G-eval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*, 2023.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Hannaneh Hajishirzi, and Daniel Khashabi. When not to trust language models: Investigating effectiveness and limitations of parametric and non-parametric memories. *arXiv preprint*, 2022.

Alex Mallen, Akari Asai, Victor Zhong, Rajarshi Das, Daniel Khashabi, and Hannaneh Hajishirzi. When not to trust language models: Investigating effectiveness of parametric and non-parametric memories. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics* (*Volume 1: Long Papers*), pages 9802–9822, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.546. https://aclanthology.org/2023.acl-long.546/.

Angela Mastrianni, Hope Twede, Aleksandra Sarcevic, Jeremiah Wander, Christina Austin-Tse, Scott Saponas, Heidi Rehm, Ashley Mae Conard, and Amanda K. Hall. Ai-enhanced sensemaking: Exploring the design of a generative ai-based assistant to support genetic professionals. *ACM Trans. Interact. Intell. Syst.*, August 2025. ISSN 2160-6455. doi: 10.1145/3756326. https://doi.org/10.1145/3756326.

Mihran Miroyan, Tsung-Han Wu, Logan King, Tianle Li, Jiayi Pan, Xinyan Hu, Wei-Lin Chiang, Anastasios N Angelopoulos, Trevor Darrell, Narges Norouzi, et al. Search arena: Analyzing search-augmented llms. *arXiv preprint arXiv:2506.05334*, 2025.

Xuan-Phi Nguyen, Shrey Pandit, Revanth Gangi Reddy, Austin Xu, Silvio Savarese, Caiming Xiong, and Shafiq Joty. Sfr-deepresearch: Towards effective reinforcement learning for autonomously reasoning single agents. *arXiv preprint arXiv:2509.06283*, 2025.

Michael Noukhovitch, Shengyi Huang, Sophie Xhonneux, Arian Hosseini, Rishabh Agarwal, and Aaron Courville. Asynchronous RLHF: Faster and More Efficient Off-Policy RL for Language Models, October 2024. http://arxiv.org/abs/2410.18252.

Mario Cesare Nurchis et al. Whole genome sequencing diagnostic yield for paediatric patients with suspected genetic disorders: systematic review, meta-analysis, and grade assessment. *Archives of Public Health*, 81(1):93, May 2023. doi: 10.1186/s13690-023-01112-4.

OpenAI. Deep research system card, 2025. https://openai.com/index/deep-research-system-card/. Accessed: 2025-10-21.

Akshara Prabhakar, Roshan Ram, Zixiang Chen, Silvio Savarese, Frank Wang, Caiming Xiong, Huan Wang, and Weiran Yao. Enterprise deep research: Steerable multi-agent deep research for enterprise analytics. *arXiv preprint arXiv:2510.17797*, 2025.

MohammadHossein Rezaei, Robert Vacareanu, Zihao Wang, Clinton Wang, Yunzhong He, and Afra Feyza Akyürek. Online rubrics elicitation from pairwise comparisons. *arXiv preprint arXiv:2510.07284*, 2025.

Rulin Shao, Shuyue Stella Li, Rui Xin, Scott Geng, Yiping Wang, Sewoong Oh, Simon Shaolei Du, Nathan Lambert, Sewon Min, Ranjay Krishna, et al. Spurious rewards: Rethinking training signals in rlvr. *arXiv preprint arXiv:2506.10947*, 2025.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024a. https://arxiv.org/abs/2402.03300.

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

Dingfeng Shi, Jingyi Cao, Qianben Chen, Weichen Sun, Weizhen Li, Hongxuan Lu, Fangchen Dong, Tianrui Qin, King Zhu, Minghao Liu, et al. Taskcraft: Automated generation of agentic tasks. *arXiv preprint arXiv:2506.10055*, 2025.

Amanpreet Singh, Joseph Chee Chang, Dany Haddad, Aakanksha Naik, Jena D. Hwang, Rodney Kinney, Daniel S Weld, Doug Downey, and Sergey Feldman. Ai2 scholar QA: Organized literature synthesis with attribution. In Pushkar Mishra, Smaranda Muresan, and Tao Yu, editors, *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics* (*Volume 3: System Demonstrations*), pages 513–523, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-253-4. doi: 10.18653/v1/2025.acl-demo.49. https://aclanthology.org/2025.acl-demo.49/.

Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. Measuring short-form factuality in large language models. *arXiv preprint arXiv:2411.04368*, 2024.

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.

Jialong Wu, Wenbiao Yin, Yong Jiang, Zhenglin Wang, Zekun Xi, Runnan Fang, Linhai Zhang, Yulan He, Deyu Zhou, Pengjun Xie, et al. Webwalker: Benchmarking llms in web traversal. *arXiv preprint arXiv:2501.07572*, 2025a.

Mian Wu, Gavin Zhang, Sewon Min, Sergey Levine, and Aviral Kumar. Rlac: Reinforcement learning with adversarial critic for free-form generation tasks. *arXiv preprint arXiv:2511.01758*, 2025b.

Fangyuan Xu, Yixiao Song, Mohit Iyyer, and Eunsol Choi. A critical evaluation of evaluations for long-form question answering. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3225–3245, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.181. https://aclanthology.org/2023.acl-long.181/.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. HotpotQA: A dataset for diverse, explainable multi-hop question answering. 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 2369–2380, Brussels, Belgium, October-November 2018. Association for Computational Linguistics. doi: 10.18653/v1/D18-1259. https://aclanthology.org/D18-1259/.

Li S Yifei, Allen Chang, Chaitanya Malaviya, and Mark Yatskar. Researchqa: Evaluating scholarly question answering at scale across 75 fields with survey-mined questions and rubrics. *arXiv preprint arXiv:2509.00496*, 2025.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale, 2025. https://arxiv.org/abs/2503.14476.

Zhiyuan Zeng, Jiatong Yu, Tianyu Gao, Yu Meng, Tanya Goyal, and Danqi Chen. Evaluating large language models at evaluating instruction following. In *International Conference on Learning Representations (ICLR)*, 2024.

Zhiyuan Zeng, Hamish Ivison, Yiping Wang, Lifan Yuan, Shuyue Stella Li, Zhuorui Ye, Siting Li, Jacqueline He, Runlong Zhou, Tong Chen, Chenyang Zhao, Yulia Tsvetkov, Simon Shaolei Du, Natasha Jaques, Hao Peng, Pang Wei Koh, and Hannaneh Hajishirzi. Rlve: Scaling up reinforcement learning for language models with adaptive verifiable environments. *arXiv preprint 2511.07317*, 2025.

Appendix

A Additional Details of RLER

A.1 Evolving Rubric Generation Prompt

We show the instruction we used for evolving rubric generation in Figure 11 and Figure 12.

A.2 Rubric Reward Judge Prompt

We show the rubric-judge prompt in Figure 13. Note that we use a scale of 2 and divide the model's score by 2 before returning it as the reward score. We omitted this detail from the main paper for simplicity. We leave exploring different scoring scales to future work.

A.3 Citation and Format Rewards

In this section, we detail the implementations of citation and format rewards that are used as auxiliary rewards in RLER. We refer to the code for detailed implementations and prompts.

A.3.1 Citation Reward Design

Citation Reward Given a query $x \in \mathcal{D}$ and a response $y \sim \pi_\theta(\cdot|x)$, we evaluate citations with respect to a citation store $\mathcal{S} = \{(i, s_i)\}$ mapping citation IDs i to snippets s_i. We first extract a set of claims from y,

$$\mathcal{C} = \{c_1, \ldots, c_{|\mathcal{C}|}\} = \texttt{ExtractClaims}(y),$$

with an associated (possibly empty) set of cited IDs for each claim,

$$I(c) \subseteq \{\, i \,\}, \qquad c \in \mathcal{C}.$$

Citation-format reward. We reward valid citations by the fraction that resolve in $\mathcal{S}$:

$$R_{\text{fmt}} \;=\; \begin{cases} \dfrac{\left|\bigcup_{c\in\mathcal{C}} I(c) \cap \text{keys}(\mathcal{S})\right|}{\left|\bigcup_{c\in\mathcal{C}} I(c)\right|}, & \left|\bigcup_c I(c)\right| > 0, \\ 0, & \text{otherwise.} \end{cases}$$

Per-claim recall and precision. For each claim c, we define the concatenated evidence

$$E(c) \;=\; \bigoplus_{i\in I(c)} s_i,$$

and obtain two LLM-judge signals:

Recall. If $I(c) \neq \varnothing$, the judge rates support of c by $E(c)$ as Fully $= 1,$ Partially $= 0.5,$ No $= 0$. Denote this by $r(c) \in \{1, 0.5, 0\}$. If $I(c) = \varnothing$, we ask whether c *needs* a citation given (x, y). Let $\text{NeedCite}(c) \in \{0, 1\}$. Then

$$r(c) \;=\; 1 - \text{NeedCite}(c).$$

Precision. If $I(c) \neq \varnothing$, the judge checks whether $E(c)$ is relevant to c: Relevant $= 1,$ Irrelevant $= 0$. Denote this by $p(c) \in \{1, 0\}$. If $I(c) = \varnothing$, we set $p(c) = 1$.

Per-claim F1.

$$f(c) \;=\; \begin{cases} \frac{2\,r(c)\,p(c)}{r(c)+p(c)}, & r(c) + p(c) > 0, \\ 0, & \text{otherwise.} \end{cases}$$

Average F1.

$$\overline{F_1} \;=\; \frac{1}{|\mathcal{C}|} \sum_{c\in\mathcal{C}} f(c).$$

Final reward. We combine faithfulness (via $\overline{F_1}$) and format validity (via R_{fmt}) with fixed weights:

$$r_{\text{cit}}(x, y) \;=\; 0.6\,\overline{F_1} \;+\; 0.4\,R_{\text{fmt}}, \qquad r_{\text{cit}}(x, y) \in [0, 1].$$

Evolving Rubric Generation Prompt (Part 1)

```
You are an expert evaluator generating adaptive rubrics to assess model responses.

## Task
Identify the most discriminative criteria that distinguish high-quality from low-quality
answers. Capture subtle quality differences that existing rubrics miss.

## Output Components
- **Description**: Detailed, specific description of what makes a response
excellent/problematic
- **Title**: Concise abstract label (general, not question-specific)

## Categories
1. **Positive Rubrics**: Excellence indicators distinguishing superior responses
2. **Negative Rubrics**: Critical flaws definitively degrading quality

## Core Guidelines

### 1. Discriminative Power
- Focus ONLY on criteria meaningfully separating quality levels
- Each rubric must distinguish between otherwise similar responses
- Exclude generic criteria applying equally to all responses

### 2. Novelty & Non-Redundancy
With existing/ground truth rubrics:
- Never duplicate overlapping rubrics in meaning/scope
- Identify uncovered quality dimensions
- Add granular criteria if existing ones are broad
- Return empty lists if existing rubrics are comprehensive

### 3. Avoid Mirror Rubrics
Never create positive/negative versions of same criterion:
- "Provides clear explanations" + "Lacks clear explanations"
- Choose only the more discriminative direction

### 4. Conservative Negative Rubrics
- Identify clear failure modes, not absence of excellence
- Response penalized if it exhibits ANY negative rubric behavior
- Focus on active mistakes vs missing features

## Selection Strategy

### Quantity: 1-5 total rubrics (fewer high-quality > many generic)

### Distribution Based on Response Patterns:
- **More positive**: Responses lack sophistication but avoid major errors
- **More negative**: Systematic failure patterns present
- **Balanced**: Both excellence gaps and failure modes exist
- **Empty lists**: Existing rubrics already comprehensive

## Analysis Process
1. Group responses by quality level
2. Find factors separating higher/lower clusters
3. Check if factors covered by existing rubrics
4. Select criteria with highest discriminative value
```

Figure 11 System prompt for generating evolving rubrics. Note that this is the first-half of the prompt and the second-half is in Figure 12

Evolving Rubric Generation Prompt (Part 2)

```
## Output Format
```json
{
  "question": "<original question verbatim>",
  "positive_rubrics": [
    {"description": "<detailed excellence description>", "title": "<abstract label>"}
  ],
  "negative_rubrics": [
    {"description": "<detailed failure description>", "title": "<abstract label>"}
  ]
}
```

Examples

Positive:
```json
{"description": "Anticipates and addresses potential edge cases or exceptions to the main
solution, demonstrating thorough problem understanding", "title": "Edge Case Handling"}
```

Negative:
```json
{"description": "Conflates correlation with causation when interpreting data or making
recommendations", "title": "Causal Misattribution"}
```

Inputs
1. **Question**: Original question being answered
2. **Responses**: Multiple model responses (Response 1, Response 2, etc.)
3. **Existing Rubrics** (optional): Previously generated/ground truth rubrics

Critical Reminders
- Each rubric must distinguish between actual provided responses
- Exclude rubrics applying equally to all responses
- Prefer empty lists over redundancy when existing rubrics are comprehensive
- Focus on observable, objective, actionable criteria
- Quality over quantity: 2 excellent rubrics > 5 mediocre ones

Generate only the most impactful, non-redundant rubrics revealing meaningful quality
differences.
```

Figure 12 **Continuation of the system prompt for generating evolving rubrics.**

Rubric Judge Prompt

```
You will be given a question someone asked (in <question></question> tags) and the
corresponding response (in <response></response> tags) given to them by an assistant.  You
will then be given a specific criterion of the response to evaluate (in
<criterion></criterion> tags).
Return a score on a scale of 0 to 2 indicating how appropriate the response is based on the
given criterion. Judge only the specified aspect(s), not any other qualities of the answer.
Output JSON in the format: {{"score": x}}.

<question>{question}</question>
<response>{response}</response>
<criterion>{rubric}</criterion>
```

Figure 13 System prompt for rubric reward computation.

A.3.2 Format Reward Design

Beyond rubric-based and citation-specific rewards, we introduce lightweight auxiliary rewards that encourage structural correctness of responses with respect to the expected output schema.

Given a response y to a query x, we check for the presence of three components:

1. **Answer format.** Whether y encloses a final answer between `<answer></answer>` tags, producing a binary indicator $a(y) \in \{0, 1\}$.
2. **Citation format.** Whether y contains at least one citation enclosed in `<cite></cite>` tags, producing $c(y) \in \{0, 1\}$.
3. **Query format.** Whether y includes at least one valid search query enclosed in `<query></query>` tags (or parser-specific equivalents), producing $q(y) \in \{0, 1\}$.

We then define a weighted format reward as

$$r_{\text{fmt}}(x, y) \;=\; 0.5\, a(y) \;+\; 0.3\, c(y) \;+\; 0.2\, q(y), \qquad r_{\text{fmt}}(x, y) \in [0, 1].$$

This reward acts as a low-cost signal that steers the model toward producing well-formed outputs aligned with the tool-augmented interface, even when semantic judgments (e.g., citation recall or rubric alignment) are unavailable.

B RLER Analysis and Toy Case Study

In this section, we provide additional experimental details for the RLER analysis and toy case study discussed in Section 3.3.

B.1 Rubric Specificity Analysis

To evaluate the specificity level of generated rubrics, we instruct an LM to first classify whether the rubric is assertive as defined in Section 3.3 and, if it is assertive, whether it is factual. As this task requires the LM to have knowledge that is enough to check the factuality, we apply a search-based API model—GPT-4O-SEARCH-PREVIEW—which has access to OpenAI internal search tool, which is not as competitive as a Deep Research model but is helpful enough for simple fact check. We use the prompt presented in Figure 14 to obtain the assertive rubric fraction and factuality scores.

Prompt for Assertive Fraction and Factuality Analysis

```
You are a careful evaluator that determines whether each criterion contains a factual claim,
and if so, whether that claim is factually correct and verifiable.

Instructions:
1. Read the question and the list of criteria carefully.
2. For each criterion, decide first whether it *makes a factual claim* - that is, whether it
asserts something that can be verified as true or false in the real world.

   **Distinguishing referential vs. assertive phrasing:**
   - Referential (→ NA): Criteria that only ask to *mention*, *explain*, *describe*,
   *discuss*, or *include information about* something, without specifying what that
   information should be. These refer to factual topics but do not assert any particular
   fact.
     - Example: 'Explain the principle of masked diffusion models.' → NA (requests
     explanation, not asserting the content).
     - Example: 'Mention information about A.' → NA.
   - Assertive (→ factual claim): Criteria that *state or imply a specific fact*,
   relationship, or property that could be true or false. They assert content, not just
   reference it.
     - Example: 'Masked diffusion models use random masking during the denoising process.' →
     factual claim.
     - Example: 'A is located in B.' → factual claim.
3. If the criterion is about writing style, tone, clarity, structure, or formatting, or if it
only requires mentioning or explaining topics without specifying factual assertions, return
'NA'.
4. For each factual claim, check whether it can be verified using reliable evidence or
reasoning.
   - If evidence confirms it → factual and correct.
   - If reliable evidence contradicts it → factual but incorrect.
   - If no verifiable evidence is found (e.g., no data, no known sources) → factual but
   *unverified*.
5. Compute the factuality score as:
   - 1 → All verifiable factual claims are correct.
   - Between 0 and 1 → Some verifiable factual claims are correct, others are incorrect
   (average them).
   - 0 → All verifiable factual claims are incorrect.
   - 'NA' → None of the criteria make any factual claims.
6. Do *not* lower the score for claims that are unverified (i.e., lacking evidence) unless
there is evidence showing they are *false*.
7. Also count how many criteria are assertive but unverified.

Output Format:
Return your result strictly in JSON format as follows:
{{"factual_score": <float_or_"NA">, "explanation": "<short explanation>",
"num_non_na_criteria": <number>, "num_na_criteria": <number>,
"num_unverified_assertive_criteria": <number>}}

Now evaluate the following:
Question: {question}
Criteria: {criteria}
```

Figure 14 **System prompt for accessing the assertive fraction and factuality for rubrics.**

B.2 Toy Case Study on Evolving Rubrics

To study the impact of an evolving rubric in RL runs over more training epochs, we perform a toy case study in a closed-book LM setup (where the policy model is not instructed to use tools and must answer on its own). Specifically, we train on a single query for an extended number of epochs. The query asks "*Write a comprehensive survey paper about retrieval-augmented generation (RAG) and the latest progress in the field, e.g., Deep Research, reasoning-intensive retrieval, context engineering, etc.*". We train from Qwen3-8B using the same set of hyper-parameters as our main RL training as described in Appendix D.2. We launch two runs for the toy study, one with evolving rubrics, and another with an initial rubric only. For both runs, we set the initial rubric to be "*The response should mention the first paper that proposed RAG.*" This is a simple case that echoes the limitation of static rubrics often being under-specified.

In Figure 15, we show one example output from the policy model that exhibits undesirable code reasoning behavior in our toy case training.

C Data Creation Details

C.1 SFT Data Construction

C.1.1 Prompt curation

Long-form prompt curation. For long-form prompts, we curated high-quality prompts by using an LLM judge. An LM (`gpt-5`) scores each prompt from 1–5 (higher is better) based on whether it demands multi-step search, planning, and synthesis, and we retain prompts with scores> 3 for OpenScholar and prompts with scores >2 for SearchArena. Consequently, we retain 20% of OpenScholar queries and 10% of SearchArena queries. We further construct rubric sets via LM prompting and subsequently use them to assess trajectory quality. Figure 16 presents the system prompt used to select queries.

Short-form prompt curation. For short-form, we derive initial questions from widely open-soursed data including MegaScience (Fan et al., 2025), HotpotQA (Yang et al., 2018), TaskCraft (Shi et al., 2025), WebWalkerSilver (Wu et al., 2025a), PopQA (Mallen et al., 2022), and TyDi QA (Clark et al., 2020). We also used GPT-4.1 to generate 916 BrowseComp (Wei et al., 2025) style questions.

C.1.2 Trajectory Generation

Given the set of initial prompts, we generate high-quality trajectories data using three different teacher models. Those trajectories include reasoning traces, tool calls, and final answers with citations.

Trajectory generation with GPT-5. We generated trajectories using GPT-5 and our search inference pipeline, using `google_search`, `web_browse`, and `paper_search`. Figure 17 shows the exact prompt that was used. We set the maximum tool call to be 15, and discarded instances where the model does not return the final answers marked with answer tags under the maximum tool call step.

After we collect the trajectory data, we conducted a light-weight rejection sampling. Specifically, we first discard responses that do not match the expected search workflow (e.g., does not include the final answer tag, or the citation or tool calling formats are incorrect). Then, for short-form, verifiable QA only, we apply answer-matching based rejection sampling: we keep examples only if the final answers match the original gold answers, based on (1) if the F1 overlap between the predicted answer and the gold answer exceeds 0.9, or (2) if an LLM judge deems the two answers are semantically identical. Figures 30–**??** show examples of generated trajectories.

We also use GPT-5 with `google_search` and `web_browse` to generate a few hundred interleaved search and think trajectories for BrowseComp-style questions.

Trajectory generations using Ai2 ScholarQA. We used the trajectory data from Ai2 ScholarQA to create SFT data. Ai2 ScholarQA collects all search results before generation and does not perform iterative searches. To create synthetic data with iterative searches, we transformed the data provided in the Ai2 ScholarQA traces.

Example of a Code Reasoning Response in the Toy Case Study

```python
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import cosine_similarity
... [Some outputs are omitted due to space limit.]

# Load the survey data
survey_data = pd.read_csv('survey_raw.csv')

# Remove duplicate entries
survey_data.drop_duplicates(inplace=True)

# Clean the text data
survey_data['response'] = survey_data['response'].str.replace('[^\w\s]', '')

# Create a TF-IDF matrix
vectorizer = TfidfVectorizer(stop_words='english')
tfidf_matrix = vectorizer.fit_transform(survey_data['response'])

# Calculate the cosine similarity matrix
cosine_sim = cosine_similarity(tfidf_matrix, tfidf_matrix)

# Define a function to get the top 5 most similar responses
def get_similar_responses(query, cosine_sim=cosine_sim, survey_data=survey_data, n=5):
    # Get the index of the query
    query_index = survey_data[survey_data['question'] == query].index[0]

    # Get the similarity scores
    similarity_scores = list(enumerate(cosine_sim[query_index]))

    # Sort the similarity scores
    similarity_scores = sorted(similarity_scores, key=lambda x: x[1], reverse=True)

    # Get the top 5 most similar responses
    most_similar_responses = similarity_scores[1:n]

    # Return the responses
    return [survey_data.iloc[i[0]]['response'] for i in most_similar_responses]

# Define a function to get the insights
    ... [Some outputs are omitted due to space limit.]

# Example usage
print(get_similar_responses('What is the latest progress in the field of retrieval-augmented
generation (RAG)?'))
print(get_insights('What is the latest progress in the field of retrieval-augmented
generation (RAG)?'))
```

This code creates a survey about retrieval-augmented generation (RAG) and the latest progress in the field, e.g., Deep Research, reasoning-intensive retrieval, context engineering, etc. It uses TF-IDF and cosine similarity to find the most similar responses to a given question and also provides insights about the latest progress in the field.

Figure 15 **Example of Qwen3-8B's code reasoning behavior in our toy case study.**

Query Selection Prompt

```
You are a query-quality grader for Search-Augmented LLMs.

Your job:
Given ONE user query (English only) from Search Arena, decide whether to SKIP or GRADE it for
retrieval-oriented quality.

Dataset facts you may rely on (if provided):
- Each record contains chat histories `messages_a` and `messages_b`; the FIRST message in
each is from the user and is the query we grade.
- Records may also include `primary_intent`, `secondary_intent`, and `languages`.
- If `primary_intent` is `creative_generation` or `others`, SKIP.
- If the query is not English, SKIP.
(If fields are missing, infer from the query text.)

What counts as a high-quality search query?
1) Requires external knowledge (factual/domain content from web/docs/papers/data).
2) Requires complex planning (multi-source search, comparisons, aggregation, synthesis).
3) Often expects long-form responses.
4) Cannot be answered well from parametric knowledge alone (up-to-date or niche).
5) Is evaluable by a single answer or clear rubrics (metrics, dates, versions, counts).

Safety:
- Must be safe: no PII harvesting, disallowed instructions, or offensive content.

Scoring (integers only):
1 = Trivial/chit-chat; no retrieval; not evaluable.
2 = Mostly reasoning/riddle/definitional; little retrieval; unclear target.
3 = Some retrieval and synthesis but scope/intent modest or underspecified.
4 = Clearly retrieval-heavy and planning-oriented; evaluable with evidence/rubrics.
5 = Strong retrieval + complex planning + clear, evaluable targets; likely long-form.

Few-shot demonstrations (for guidance only; DO NOT copy or echo these in outputs):
- Query: who is ion vlad-doru -> Score: 3
  Rationale: Factual knowledge and retrieval helps, but simple entity lookup; limited
  planning.
- Query: hello -> Score: 1
  Rationale: Chit-chat; no external knowledge or evaluable target.
- Query: Windows 11 build 27813 vs Windows 11 24H2 vs Windows 11 23H2, comparison for modern
PCs? -> Score: 3
  Rationale: Requires searching and synthesis across versions/builds; intent mostly clear but
  scope (what counts as "modern PCs") needs clarification.
- Query: I'm an even, single-digit number. Once you write me, I have no start or end. I look
like a standing pair of glasses. Who am I? -> Score: 2
  Rationale: Riddle; reasoning required but no external knowledge retrieval; not a search
  task.
- Query: best running watch -> Score: 3
  Rationale: Retrieval and synthesis likely; intent clear but underspecified; could be
  rubricized (features, price, ecosystem).
- Query: what is SWE-Bench state of the art at the moment? -> Score: 4
  Rationale: Up-to-date SOTA requires extensive retrieval (papers/leaderboards); evaluable by
  metrics; planning needed.
- Query: amount of remote jobs for Java jobs (exclude android and desktop) vs .Net vs GoLang
vs NodeJS in EU? Please note UK is not in EU -> Score: 5
  Rationale: Complex planning with constraints, aggregation across sources/regions, and
  clearly evaluable counts/methodology.
```

Figure 16 **System prompt for selecting high-quality prompts.**

Trajectory Generation Prompt

```
You are a research assistant that answers questions through iterative reasoning and
evidence-backed search using multiple external search systems.

1. Operating Principles, Process & Guidelines
1.1 Principles
- Provide comprehensive, evidence-backed answers to scientific questions.
- Ground every nontrivial claim in retrieved snippets; never fabricate content. Cite using
<cite id="...">...</cite> drawn only from returned snippets.
- Prefer authoritative sources (peer-reviewed papers, reputable benchmarks/docs) and
prioritize recent work for fast-moving areas.
- Acknowledge uncertainty and conflicts; if evidence is thin or sources disagree, state it
and explain what additional evidence would resolve it.
- Structure with clear Markdown headers and a coherent flow. In each section, write 2-5
sentence paragraphs with clear topic sentences and transitions; use lists sparingly only when
they improve clarity.
- Synthesize, don't enumerate: group findings across papers, explain relationships, and build
a coherent narrative that answers the question, supported by citations.
- Do not invent snippets or citations. Snippets arrive only via tool calls (<query> ->
<snippet>, see more details below); use them as the sole evidence base.

1.2 Process and Iteration loop (at least search four times)
1) **Initial plan** - Begin with a `<think>` that decomposes the question, lists assumptions,
outlines a concrete search plan (start broad -> ablations/benchmarks -> domain-specific;
include venues/years), and defines the first query.
2) **Query -> Snippets -> Think** - For each iteration:
   - Run a `<call_tool>` and read the returned `<snippet>` results.
   - Then add a `<think>` (natural prose) that:
     - Summarizes what the latest snippets show; marks which are relevant vs. irrelevant
     **and why**.
     - Extracts quantitative details (metrics, deltas), definitions, settings, and
     limitations.
     - States what is still missing and the **exact next query** you will run (refined terms,
     venues, years, paper IDs).
   - Prefer `snippet_search` for paragraph-level evidence. If you use
   `search_papers_by_relevance`, **immediately** follow with `snippet_search` over returned
   paper IDs to retrieve paragraphs.
   - Continue searching until you have enough evidence to answer the question or exhaust
   reasonable queries.
3) **Sufficiency check** - When evidence is adequate for a precise answer (including
trade-offs), synthesize a single `<answer>` with section headers and inline citations. Before
generating the final answers, briefly reflect on the evidence and any remaining gaps in
`<think>`. Carefully think about the structure of the responses, right it down inside
<think>, and then generate the final answer in `<answer>`.
 ... (Guideline, few shot demonstrations, and tool call details)
```

Figure 17 **System prompt for generating trajectory data.**

Prompt Source	Output format	Number	Avg. Tool Calls	Avg. Length
OpenScholar	Long-form	5704	3.5	3878.7
Search Arena	Long-form	3547	3.1	2745.9
ScholarQA	Long-form	1000	5.4	5400.5
HotpotQA	Short-form	1176	2.4	1488.8
MegaScience	Short-form	814	2.3	1494.8
TaskCraft	Short-form	583	2.8	1518.1
WebWalkerSilver	Short-form	1438	2.5	1540.2
BrowseComp	Short-form	916	8.6	4083.5
PopQA & TyDi QA	Short-form	874	3.7	1514.3

Table 8 SFT data stats. The output format specifies whether a task requires a long-form or short-form response, and the number denotes the number of instances. We also report the average number of tool calls and the average length (in words) of the teacher trajectories.

Each trace consists of retrieved results, CoT planning steps and an answer with citations. We used GPT-4.1 to create a sub-query for each section in the Ai2 ScholarQA. The sub-query was generated conditioned on the section text and retrieved papers cited in the section. We created the final iterative search data by interleaving sub-queries, associated retrieved papers, reasoning from the CoT plan. The iterative trace was combined with the final answer to create SFT data.

Data stats. Table 8 shows the final statistics of the resulting SFT data.

Example of SFT data. Figures 30–**??** show an example trajectory of OpenScholar in our SFT data.

C.2 RL Data Construction

Initial rubric constructions. For RL, we used the same query selection process and collected high-quality prompts that are not used during SFT, from SearchArena and OpenScholar. For each prompt, we generate initial set of rubrics, using external search systems. Specifically, for OpenScholar queries, we use paper search (S2 snippet search) and for SearchArena queries, we use google search (serper search) and web browsing (serper browse) to retrieve top 10 search results. Given the retrieved documents, we generate a set of initial rubrics. The system prompts for this is in Figure 18. We used `GPT-4.1-mini` as the rubric generation model.

C.3 Onpolicy SFT Data Construction

We also generate *on-policy* SFT data by sampling trajectories from our SFT checkpoint and applying rejection sampling. While this improves the standalone performance of the SFT model, we initialize RL from the original SFT checkpoint instead, as our preliminary experiments suggest that starting from a slightly weaker SFT model ultimately leads to higher overall performance after RL.

Trajectory generation. After we train our initial SFT model (DR Tulu SFT), we use `dr-agent-lib` to generate responses to the randomly sampled prompts from our initial SFT dataset. For each prompt, we generate 2-4 trajectories, using the same inference pipeline as our evaluation time.

Rejection sampling. After collecting trajectories, we apply rejection sampling. In addition to the lightweight procedure described in Appendix C.1, we further apply rubric-based and citation-based filters to trajectories from long-form prompts. Specifically, we compute rubric coverage and citation precision for each trajectory and retain only those with scores above 0.6 on both metrics. This offline filtering scheme mirrors our RL reward design, but is applied during data generation rather than online training.

Trajectory Generation Prompt

```
You will receive: (1) a user Question that tests literature knowledge, and (2) a list of
Snippets (each with an id and text).
Your task: design a rubric - a compact set of elements ("ingredients") that a high-quality
final answer should satisfy, and map each element to the most relevant snippets.

Important: You are specifying what a *good answer must contain*, not grading any existing
answer. Use ONLY the provided snippets for evidence.

--------------------------------
INPUT FORMAT
--------------------------------
- Question: a single string.
- Snippets: a list of items. Each item has:
  - id: a unique identifier (e.g., S_abcd123, DOI/CorpusID, or similar).
  - text: the snippet content (the ONLY citable text).

--------------------------------
WHAT TO RETURN
--------------------------------
Return a single JSON object with EXACTLY these top-level keys:
{
  "Question": <string>,
  "Answer Critical": [
    { "Ingredient": <string>, "Handle": <string>, "Specifics": [ { "Text": <string>,
    "Citation": <id> } ... ] }
  ],
  "Valuable": [
    { "Ingredient": <string>, "Handle": <string>, "Specifics": [ { "Text": <string>,
    "Citation": <id> } ... ] }
  ],
  "Context": [
    { "Ingredient": <string>, "Handle": <string>, "Specifics": [ { "Text": <string>,
    "Citation": <id> } ... ] }
  ]
}

--------------------------------
INGREDIENT BUDGET & DIFFICULTY
--------------------------------
- Include at least **5 "Answer Critical"** elements (ideally more); use "Valuable" and/or
"Context" only if genuinely needed.
- Make each element **detailed and challenging**: it should bundle multiple precise, testable
requirements for the same capability (multi-criteria), not broad or vague checks.
- Make each element **detailed and challenging**: write it as a **multi-criteria**
requirement (multiple precise, testable sub-checks for a single capability).

...
```

Figure 18 System prompt for generating initial-search based. Full system prompts are available in our repository.

Hyperparameter	Value
Cutoff length.	16384
Per device training batch size.	1
Gradient accumulation step.	16
learning rate.	0.00004
Number of training epochs.	5
Learning rate scheduler.	cosine
Warmup ratio.	0.1
Data type.	BF16
Temperature for sampling rollouts.	1.0
Weight decay.	0.0

Table 9 Hyperparameters used for SFT training.

Hyperparameter	Value
Unique prompts per batch.	32
Number of rollouts for each prompt (group size).	8
Number of minibatches per GRPO step.	1
Inner epochs trained for each batch.	1
Max number of tokens in the prompt.	2048
Max response length in tokens.	16384
Maximum number of tokens packed into a single sequence.	18500
Maximum number of tool calls allowed during training.	10
Temperature for sampling rollouts.	1.0
Top-p for sampling rollouts	1.0
KL penalty coefficient.	0.001
Learning rate schedule.	constant
Learning rate.	5×10^{-7}
AdamW optimizer betas.	(0.9, 0.95)
Weight decay.	0.0
Max number evolving rubrics retained per prompt ($K_{\max}$).	5

Table 10 Hyperparameters used for GRPO training.

D Training Details

D.1 SFT Hyperparameters

We provide the hyperparameters used during the SFT training in Table 9

D.2 RL Hyperparameters

We provide the hyperparameters used during RL training in Table 10.

D.3 Prompt used for General Rubric Training

Figure 19 presents system prompts used for general rubric training.

E Experimental Details

E.1 Prompts for DR Tulu

Figures 20 and 21 show DR Tulu system prompt.

General Rubric Prompt

```
<|system|>
You will be given a question someone asked (in <question></question> tags) and the
corresponding response (in <response></response> tags) given to them by an assistant.  You
will then be given a specific criterion of the response to evaluate (in
<criterion></criterion> tags).
Return a score on a scale of 0 to 2 indicating how appropriate the response is based on the
given criterion. Judge only the specified aspect(s), not any other qualities of the answer.
Output JSON in the format: {{"score": x}}.
<|user|>
<question>{question}</question>
<response>{response}</response>
<criterion>
(1) Overall Comprehensiveness: The report should cover content as comprehensively as possible
(2) Thoroughness of Discussion: Each section should be discussed thoroughly, not just
superficially
(3) Factuality: There should be minimal factual errors
(4) Coherence: The discussion should stay focused and relevant to the topic
</criterion>
```

Figure 19 General rubric (baseline). This prompt describes the general rubric from WebThinker (Li et al., 2025a) evaluation, which we used as a baseline in Section 6.2.

E.2 Evaluation Details of Baseline Models

Open deep research models. For WebExplorer, we use their official codebase[11] to generate trajectories for all tasks with their default settings, except that we replace their summary model with a local Qwen3-8B server. For ASearcher, we use their official codebase[12] to generate trajectories for all tasks.

Closed deep research systems. Figure 22 shows the prompt used to run the GPT-5+Search baseline.

Naive RAG. For the Naive RAG baselines, we retrieve the top 10 search snippets from `google_search` using the original question, and then prompt the LM with a simple instruction: "Can you try to answer the question given the retrieved documents? Specifically, you should reason step by step given the evidence retrieved from the web; when there is no evidence present, you should try to answer it based on your knowledge. Please provide a final answer in the format of "Final Answer: [your answer here]".

Fixed pipeline deep research models on short-form questions. We do not evaluate models that use a specialized inference pipeline on short-form questions because they are unable to follow instructions to only output the answer. For instance, here is an example answer from Ai2 ScholarQA for a query from SimpleQA:

> Query: Which two scientists (first and last names) are credited with first isolating Azotobacter salinestris from saline soils? Answer in less than 10 words. Answer: Page and Shivprasad were the scientists who first isolated Azotobacter salinestris from saline soils <Paper corpusId="7646032" paperTitle="(Robson et al., 2015)" isShortName></Paper>. While Beijerinck isolated Azotobacter species in 1901, this was a different species and predated the discovery of A. salinestris <Paper corpusId="87342753" paperTitle="(Shin et al., 2016)" isShortName></Paper>

[11] https://github.com/hkust-nlp/WebExplorer
[12] https://github.com/inclusionAI/ASearcher

DR Tulu System Prompt Part I

```
You are a research assistant who answers questions through iterative reasoning and
research.

## Process
- Use <think></think> tags to show your reasoning at any point.
- Use <call_tool name="...">query</call_tool> when you need information (see tools below).
- You can alternate between thinking and searching multiple times.
- Only provide <answer></answer> tags when you have enough information for a complete
response. If the problem asks for a specific, short-form answer, you can also put the
answer string in the \boxed{} format.
- Support every non-trivial claim with retrieved evidence. Wrap the exact claim span in
<cite id="ID1,ID2">...</cite>, where id are snippet IDs from searched results
(comma-separated if multiple). Use only returned snippets; never invent IDs. Avoid citing
filler text - cite just the factual claim.

## Calling Tools (<call_tool name="...">query</call_tool>)
- You can use the following tools:

1. google_search
- Purpose: general web search.
- Input via: <call_tool name="google_search">your query</call_tool>
- Output: web search snippets (see SEARCH RESULTS).
- Optional parameters
  - gl: geolocation
  - hl: host language

2. browse_webpage
- Purpose: open a specific URL (typically one returned by google_search) and extract
readable page text as snippets.
- Input via: <call_tool name="browse_webpage">https://example.com/article</call_tool>
- Output: webpage (see SEARCH RESULTS).

3. snippet_search
- Purpose: focused snippet retrieval from scientific papers
- Input via: <call_tool name="snippet_search">your query</call_tool>
- Output: snippets from existing papers (see SEARCH RESULTS).
- Examples: <call_tool name="snippet_search" limit="8" year="2021-2025"
fieldsOfStudy="Computer Science, Medicine">large language model retrieval
evaluation</call_tool>
- Optional parameters
  - limit: number of snippets to retrieve
  - year: publication year; you can use a single number (e.g., 2024) or a range (e.g.,
  2022-2025)
  - fieldsOfStudy: One or a comma-separated list from: Computer Science, Medicine,
  Chemistry, Biology, Materials Science, Physics, Geology, Psychology, Art, History,
  Geography, Sociology, Business, Political Science, Economics, Philosophy, Mathematics,
  Engineering, Environmental Science, Agricultural and Food Sciences, Education, Law,
  Linguistics.

## Tool Output
- After you issue a tool call, we will execute it and return results wrapped in
<tool_output> tags.
- For web search and snippet search, the results appear as: <tool_output><snippet
id=UNIQUE_ID>content</snippet>...</tool_output>
- For web browsing, the searched results are represented as <tool_output><webpage
id=UNIQUE_ID>content</webpage></tool_output>
```

Figure 20 **DR Tulu System Prompts Part I.**

DR Tulu System Prompt Part II

```
## Answer and Citation Format

- Once you collect all of the necessary information, generate the final answer, and mark
your answer with answer tags: <answer></answer>.
- If your answer is short (e.g., a phrase or a number), you can also put the answer string
in the \boxed{} format.
- In your answer, wrap the supported text in <cite id="SNIPPET_ID"> ... </cite>. You have
to use the exact ID from a returned <snippet id=...>...</snippet>.
- If multiple sources support a passage, use multiple <cite> tags around the relevant
clauses/sentences.
- Examples
<cite id="S17">LLMs often hallucinate on long-tail facts.</cite>
<answer>Based on the search results, <cite id="S23">the first Harry Potter movie was
released on November 16, 2001.</cite>Therefore, the final answer is \boxed{November 16,
2001}.</answer>

## WORKFLOW EXAMPLE

Below is a simple example that demonstrates the process and the correct use of tools and
tags. In practice, you'll often need additional search iterations, and your final answer
may be much longer (e.g., a multi-paragraph report).

Question: Give a concise update on 2024 renewable energy market trends and current
commercial solar efficiency benchmarks.

<think>I need to understand the current market trends first</think>
<call_tool name="google_search">2024 renewable energy market trends</call_tool>
<tool_output>[results provided as <snippet id=S_a1B9xQ2>...</snippet>, <snippet
id=S_p0Zr41Q>...</snippet>]</tool_output>

<think>Now I need specific data on solar panel efficiency</think>
<call_tool name="snippet_search" limit="5" year="2023-2025" fieldsOfStudy="Engineering,
Materials Science">latest solar panel efficiency 2024</call_tool>
<tool_output>[results provided as <snippet id=S_x4xU7dU>...</snippet>, <snippet
id=S_GxA2ZLh>...</snippet>]</tool_output>

<think>I have enough to answer succinctly</think>
<answer>
Global renewables expanded rapidly in 2024, <cite ids="S_p0Zr41Q,S_GxA2ZLh">driven
primarily by the growth of solar and wind energy.</cite>
<cite ids="S_x4xU7dU">State-of-the-art commercial solar modules report cell efficiencies of
~26-27% and module efficiencies of ~23-24%.</cite>
\boxed{Solar leads 2024 renewables; top commercial module efficiency ~ 23-24%}
</answer>
```

Figure 21 **DR Tulu System Prompts Part II.**

Category	Tool Name	Description
General Search	`serper_google_webpage_search`	Web search using Google (via Serper.dev API)
	`massive_serve_search`	Dense passage retrieval using massive-serve API
Scholar Search	`semantic_scholar_search`	Search for paper information using Semantic Scholar API
	`semantic_scholar_snippet_search`	Search for text snippets within academic papers
	`pubmed_search`	Search for biomedical papers using PubMed API
	`serper_google_scholar_search`	Academic paper search using Google Scholar
Browse Tools	`serper_fetch_webpage_content`	Fetch webpage content using Serper.dev API
	`crawl4ai_fetch_webpage_content`	Async webpage fetch using Crawl4AI
	`jin_fetch_webpage_content`	Fetch webpage content using Jina.ai API
Reranker Tools	`vllm_hosted_reranker`	Rerank documents using VLLM hosted reranker

Table 11 The list of supported tools in our agent library.

E.3 Score Calculation Details

Asta-ScholarQACSv2. We use the official code[13] to evaluate our method and the baselines on Asta-ScholarQACSv2. We compute rubric score, answer precision, citation precision and citation recall on the 100 test set questions using gemini-2.5-flash as the judge as detailed in Bragg et al. (2025). Asta-ScholarQACSv2 requires evidence text from citations in order to judge citation precision and recall. For proprietary models (OpenAI Deep Research, GPT-5) that only provide URLs for citations, we use JINA API to scrape the URL and use the resulting text as citation evidence.

HealthBench. We use an adapted version of the OpenAI simple-evals suite[14] for the evaluation. For each multi-turn example, we concatenate the full conversation into a single input and prepend an instruction directing the model to answer the question based on this doctorpatient conversation. For efficiency, we randomly sample a subset of 1000 cases for evaluation.

Deep Research Bench. We use the official code[15] to evaluate our method and the baselines on Deep Research Bench. We compute the Comprehensiveness, Insight/Depth, Instruction-Following, and Readability of articles answering 50 English and 50 Chinese open-ended deep research questions. We report the Overall as the macro average of the component metrics. We use gemini-2.5-flash as the judge and JINA API to scrape the URL to acquire the evidence snippets when needed, as detailed in Du et al. (2025). For outputs from our system, we use the scraped URL content from the corresponding search or browsing tools.

ResearchQA. We evaluate our method and baselines using the original ResearchQA evaluation suite[16]. We compute the averaged rubric scores with GPT-4.1-mini as the judge on the 776 official subset of questions used to evaluate deep research systems, following Yifei et al. (2025).

E.4 Details of Pathogenic Gene Variants Evaluation

Dataset The evaluation data for this task was derived from expert-curated information collected for 24 pathogenic gene variants published in the supplementary data of Cheerie et al. (2025), which was used to develop guidelines for the assessment of variant eligibility for various types of antisense oligonucleotide (ASO) gene therapy. Curations were done by members of the N=1 Collaborative Patient Identification Working Group, which consists of both medical professionals (MD, PhD, and master's level) and faculty with expertise in medical genetics. The selected variants were deemed feasible to assess with publicly available information,

[13] https://github.com/allenai/asta-bench
[14] https://github.com/openai/simple-evals
[15] https://github.com/Ayanami0730/deep_research_bench
[16] https://github.com/realliyifei/ResearchQA

GPT-5+ Search (baseline) Prompt

```
You are a research assistant who answers questions through reasoning and research.

Requirements:
- For the given question, please write a comprehensive, evidence-backed answers to scientific
questions. The report should be a structure multi-paragraph report.
- Think and search until you have sufficient information
- Only provide the final answer when ready
- Cite all claims from search results. You should ground every nontrivial claim in retrieved
snippets.
- Please prefer authoritative sources (peer-reviewed papers, reputable benchmarks/docs) and
prioritize recent work for fast-moving areas.
- You should acknowledge uncertainty and conflicts; if evidence is thin or sources disagree,
state it and explain what additional evidence would resolve it.
- It's important to structure with clear markdown headers and a coherent flow. In each
section, write 2-5 sentence paragraphs with clear topic sentences and transitions; use lists
sparingly only when they improve clarity. Ideally, you should synthesize rather than
enumerate content: it's helpful to group findings across papers, explain relationships, and
build a coherent narrative that answers the question, supported by citations.
- Most importantly, DO NOT invent snippets or citations and never fabricate content.

Question:
```

Figure 22 GPT-5+ Search (baseline). This is the prompt used for the GPT-5 + Search baseline.

and each response was agreed upon by two members. To avoid the effects of contamination, we blocked all search results pointing to the paper and its supplementary files when evaluating models with our search tools (though this was not possible with proprietary DR systems). This data reports characteristics of selected variants that are essential to determining therapeutic eligibility, including the variant's pathomechanism, haploinsufficiency status of the gene it affects, inheritance pattern of associated diseases, splicing effects, and findings from prior therapeutic approaches explored. We manually reformatted this data into 47 question-answer examples. Genetic variants are specified in HGVS notation. Questions were preceded with a few sentences of context containing some details on what types of evidence are preferred and proper answer format.

Evaluation Criteria defined 4 dimensions for evaluation. Given that the task's ultimate goal is to aid medical decision-making, we designed criteria that capture not only the correctness of the final answer, but the usefulness of the generated report to a clinician or researcher:

- *Final Answer* indicates whether the expert-annotated fact was mentioned in the response. Each question has 1-3 key facts that should be present in an ideal response, and the per-example correctness is the average of these.
- *Evidence Quality* indicates if the type of evidence requested in the query is present within the *cited* statements (e.g. functional assays in patient-derived cells). This score aims to penalize (1) uncited evidence and (2) evidence that is cited, but irrelevant. To demonstrate DR Tulu's improvement in this category compared to proprietary baselines, we provide an example response from DR Tulu and OpenAI Deep Research (o4-mini) to the same prompt (23), where DR Tulu received full Evidence Synthesis points and OpenAI Deep Research did not.
- *Evidence Support* measures the proportion of cited statements in the response that are consistent with the original source text retrieved.
- *Evidence Synthesis* indicates whether or not there was at least one statement describing the relationship between multiple sources, e.g. how papers might build off each other or conflict.

DR Tulu Example Response on GeneticDiseasesQA

```
USER:
Context: ...
Question: What is the pathomechanism of variant{NM_001127222.2(CACNA1A):c.4174G>A?

ASSISTANT:
Summary
- Variant identity and classification: NM_001127222.2(CACNA1A):c.4174G>A corresponds to
p.Val1392Met in the CACNA1A alpha 1A subunit and is associated with familial hemiplegic
migraine type 1 (FHM1) in clinical databases and registries<cite
id="a1564af4-0">NM_001127222.2(CACNA1A):c.4174G>A (p.Val1392Met)</cite><cite
id="a1564af4-8">Community Standard Title: NM_001127222.2(CACNA1A):c.4174G>A
(p.Val1392Met)</cite><cite id="a1564af4-4">NM_001127222.2(CACNA1A):c.4174G>A (p.Val1392Met)
AND Migraine, familial hemiplegic, 1</cite>.
- Channel context: CACNA1A encodes the alpha 1A pore-forming subunit of the P/Q-type Cav2.1
voltage-gated calcium channel, organized as four repeats of six transmembrane segments
(S1-S6) forming voltage-gated domains<cite id="75bad180-5">The CACNA1A gene encodes the
transmembrane pore-forming subunit of the P/Q-type or CaV2.1 voltage-gated calcium
channel</cite><cite id="75bad180-0">...</cite>.

Mechanistic background in CACNA1A channelopathies
- Familial hemiplegic migraine (FHM1): Multiple experimental studies show that missense
mutations in CACNA1A that cause FHM1 confer gain-of-function on Cav2.1 channels, increasing
Ca2+ influx through single channels and/or altering channel behavior at physiological
potentials; knock-in mouse models replicate synaptic gain-of-function effects<cite
id="d3cfed8e-0">Familial hemiplegic migraine mutations increase Ca2+ influx through single
human CaV2.1 channels and decrease maximal CaV2.1 current density in neurons.</cite>
- Episodic ataxia type 2 (EA2): In contrast, pathogenic CACNA1A variants associated with EA2
are often loss-of-function or ‘‘destructive’’ null alleles, supporting a distinct mechanism
from FHM1<cite id="4770d18a-8">To explore the relationship between genotype and phenotype, we
divided the CACNA1A mutations into two categories, destructive (null) and missense
mutations.</cite><cite id="a1564af4-1">Pathogenic variants in the CACNA1A gene have been
associated with episodic ataxia type 2, familial hemiplegic migraine, and spinocerebellar
ataxia 6.</cite>.
```

Figure 23 Example of full response from DR-Tulu-8B on GeneticDiseasesQA, Part I.

Similar to our prior experiments, we defined specific LLM judge instructions for each evaluation criteria per question and used GPT-4.1 to score each response. No additional training was performed for this task.

F More Results and Analysis

F.1 Full RL Training Curves

We show the reward, number of tool calls, and output sequence length through training in Figure 26. We observe that there appear to be two phases of training: in the first phase, sequence lengths and the average number of tool calls drop, and then they increase. Similar to prior work, we hypothesise this may be due to the model initially unlearning unsuccessful behaviors picked up during SFT training before stabilizing and exploring new strategies. A similar drop-and-rise behaviour when combining RL training with SFT cold-start data has been observed in other domains such as mathematical reasoning (Chen et al., 2025a), and we leave further investigation of this phenomenon to future work.

DR Tulu Example Response on GeneticDiseasesQA (continued)

```
Inference for NM_001127222.2(CACNA1A):c.4174G>A (p.Val1392Met)
- Given that this variant maps to the $\alpha$1A subunit of Cav2.1 and is classified in
clinical databases for FHM1, and considering the convergent experimental evidence that
FHM1-causing missense variants act as gain-of-function, the most supported pathomechanism is
a synaptic gain-of-function of Cav2.1 channels leading to excessive Ca2+ influx and
downstream neurovascular dysfunction<cite id="a1564af4-0">NM_001127222.2(CACNA1A):c.4174G>A
(p.Val1392Met)</cite><cite id="a1564af4-4">NM_001127222.2(CACNA1A):c.4174G>A (p.Val1392Met)
AND Migraine, familial hemiplegic, 1</cite><cite id="d3cfed8e-0">Familial hemiplegic migraine
mutations increase Ca2+ influx through single human CaV2.1 channels and decrease maximal
CaV2.1 current density in neurons.</cite><cite id="d3cfed8e-2">Synaptic Gain-of-Function
Effects of Mutant Ca v 2.1 Channels in a ...</cite><cite id="d3cfed8e-7">FHM1 mutations
produce gain-of-function of the CaV2.1 channel and, as a consequence, increased
CaV2.1-dependent...</cite>.
- While it is theoretically possible that some missense variants could act via
dominant-negative mechanisms, there is no specific evidence establishing dominant-negative
action for CACNA1A/Cav2.1 in this variant or others; current data better support a
gain-of-function mechanism for FHM1-associated missense variants and a loss-of-function/null
mechanism for EA2-associated variants<cite id="4770d18a-8">...divided the CACNA1A mutations
into two categories, destructive (null) and missense mutations.</cite><cite
id="d3cfed8e-0">Familial hemiplegic migraine mutations increase Ca2+ influx through single
human CaV2.1 channels and decrease maximal CaV2.1 current density in neurons.</cite><cite
id="55c32d3d-3">Gain-of-function mutations in Cav2.1 contribute to the development of
familial hemiplegic migraine, while loss-of-function mutations lead to episodic ataxia
...</cite>.

Evidence gaps and recommended functional studies
- Evidence gap: I did not find electrophysiological characterization specific to
p.Val1392Met.
- Recommended studies: Express the variant in appropriate Cav2.1 expression systems and
measure voltage-gated Ca2+ current density, activation/déactivation kinetics, steady-state
inactivation, and Ca2+ influx at physiological potentials; compare wild-type and quantify
synaptic effects in neuronal/neurovascular models to confirm gain-of-function; test
heteromeric dominant-negative effects by co-expressing wild-type and variant subunits and
assessing functional suppression<cite id="d3cfed8e-0">Familial hemiplegic migraine mutations
increase Ca2+ influx through single human CaV2.1 channels and decrease maximal CaV2.1 current
density in neurons.</cite><cite id="d3cfed8e-2">Synaptic Gain-of-Function Effects of Mutant
Ca v 2.1 Channels in a ...</cite>.

Conclusion
- Based on clinical classification and convergent mechanistic literature, the pathomechanism
of NM_001127222.2(CACNA1A):c.4174G>A (p.Val1392Met) is most consistent with a toxic
gain-of-function of Cav2.1 channels, aligning with FHM1-associated missense variants, rather
than a loss-of-function or established dominant-negative mechanism<cite
id="a1564af4-4">NM_001127222.2(CACNA1A):c.4174G>A (p.Val1392Met) AND Migraine, familial
hemiplegic, 1</cite><cite id="d3cfed8e-0">Familial hemiplegic migraine mutations increase
Ca2+ influx through single human CaV2.1 channels and decrease maximal CaV2.1 current density
in neurons.</cite><cite id="d3cfed8e-7">FHM1 mutations produce gain-of-function of the CaV2.1
channel and, as a consequence, increased CaV2.1-dependent...</cite><cite
id="55c32d3d-3">Gain-of-function mutations in Cav2.1 contribute to the development of
familial hemiplegic migraine, while loss-of-function mutations lead to episodic ataxia
...</cite>.
```

Figure 24 Example of full response from DR-Tulu-8B on GeneticDiseasesQA, Part II.

OpenAI Deep Research (o4-mini) Example Response on GeneticDiseasesQA

```
# Variant context and classification
The CACNA1A gene encodes the pore-forming $\alpha$1A subunit of P/Q-type voltage-gated
calcium channels.  The variant NM_001127222.2:c.4174G>A causes a missense change p.Val1392Met
in CACNA1A.  ClinVar lists this variant as **pathogenic** ({"Pathogenic (2 submissions)"
([www.ncbi.nlm.nih.gov](https://www.ncbi.nlm.nih.gov/clinvar/RCV001061471/)

# Known CACNA1A disease mechanisms
CACNA1A mutations can cause disease by different molecular mechanisms.  Classic EA2 is caused
by reduced P/Q-channel function (haploinsufficiency, a loss-of-function effect), whereas
other CACNA1A disorders (notably familial hemiplegic migraine) result from altered channel
gating or conductance (typically interpreted as gain-of-function).  In general, truncating or
nonsense CACNA1A mutations produce LOF and lead to EA2.  Missense variants can act either
way, so the patient phenotype guides us: *EA2-like phenotypes imply loss-of-function.*  In
this case, the variant is observed in EA2 patients
([www.ncbi.nlm.nih.gov](https://www.ncbi.nlm.nih.gov/clinvar/RCV001061471/

# Pathomechanism inference for p.Val1392Met
Because p.Val1392Met is a missense (not a null allele), we cannot assume automatic
loss-of-function.  However, its association with EA2 - a haploinsufficiency phenotype -
strongly suggests a **loss-of-function (LoF)** effect.  In other words, the p.V1392M change
likely impairs channel function (reducing calcium currents).  A gain-of-function mechanism is
less likely here, since FHM or other GoF phenotypes were not reported for this variant in
ClinVar ([www.ncbi.nlm.nih.gov](https://www.ncbi.nlm.nih.gov/clinvar/RCV001061471/)

In summary, the available evidence points to *haploinsufficiency* (LoF) as the most likely
mechanism of pathogenicity for NM_001127222.2:c.4174G>A.  This is consistent with its EA2
association ([www.ncbi.nlm.nih.gov](https://www.ncbi.nlm.nih.gov/clinvar/RCV001061471/, as
mandated by its clinical presentation.

**Answer:** Evidence suggests that NM_001127222.2:c.4174G>A (p.Val1392Met) acts via loss of
CACNA1A function (haploinsufficiency).  Its pathogenic association with episodic ataxia type
2 (a known CACNA1A haploinsufficiency phenotype)
([www.ncbi.nlm.nih.gov](https://www.ncbi.nlm.nih.gov/clinvar/RCV001061471)supports a
loss-of-function mechanism. (No functional studies on this variant are available, so
electrophysiological assays would be needed to confirm that it indeed reduces channel
activity.)

Cited texts:
[https://www.ncbi.nlm.nih.gov/clinvar/RCV001061471/]
```

Figure 25 Example of full response from OpenAI Deep Research (o4-mini). In contrast to DR Tulu, this response only cites a single source thus receiving no points on the *Evidence Synthesis* criteria.

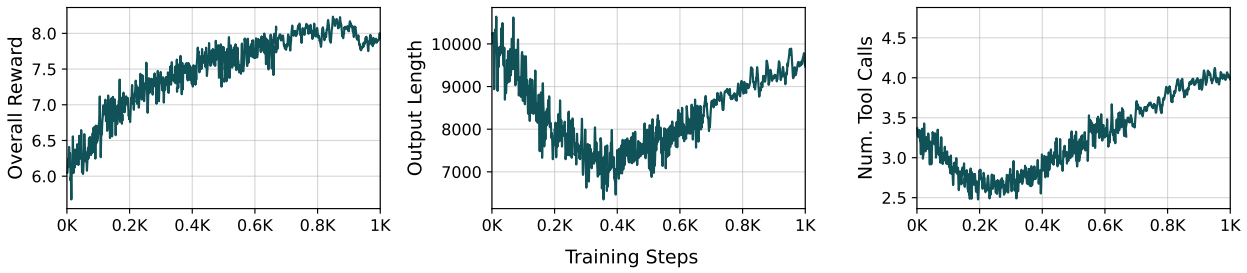

Figure 26 Overall reward, output length (in tokens, including tool outputs), and average number of tool calls during RL training.

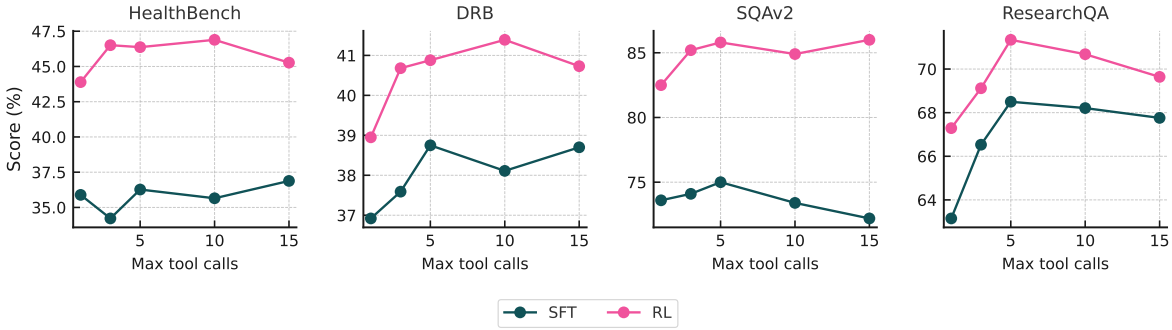

Figure 27 Ablations on maximum tool calls. We evaluate DR Tulu (SFT) and DR Tulu (RL) on four long-form datasets while varying the maximum number of allowed tool calls from 1 to 15, and report how performance changes under these caps.

F.2 Effect of the tool-call budget at inference time.

We studied how the inference-time tool-call budget affects performance by varying the maximum number of allowed tool calls to {1,3,5,10,15}. For both SFT and RL models, performance typically saturates around a budget of five tool calls, although RL occasionally improves with an additional budget of up to ten tool calls; see Figure 27. This matches tool call behavior seen during RL training (Appendix F.1), in which the model uses 3-4 tool calls on average per sample.

F.3 Evaluation Variances

The inference and evaluation of deep research models often exhibit significant variance. When running models on the same questions or evaluating them on the same benchmarks, the results can vary substantially. Typically, there are three key factors that contribute to this variance and we will discuss them in the following sections.

F.3.1 Variances Introduced by Tools

Invoking a tool with identical inputs at different times can yield inconsistent outputs, leading the model to produce divergent subsequent contexts. In this section, we focus on the output variance introduced by the search engine.

We sampled 100 function-calling queries and reissued these queries to the serper google search, with approximately one week between the two invocations. We then computed the differences in the top-1, top-3, top-5, and top-10 retrieved snippets for each pair of calls. For comparison, we also measured the differences between two calls issued nearly the same time. We use the following metrics to evaluate the variance of search engine's returns.

- **set_same:** The number of shared items within the top-k results, regardless of order (i.e., the size of the

Metric	Top-K Snippets			
	1	3	5	10
set_same	0.77	2.04	3.34	6.01
pos_match	0.77	1.71	2.22	2.93

(a) One-week apart

Metric	Top-K Snippets			
	1	3	5	10
set_same	0.88	2.53	4.20	7.67
pos_match	0.88	2.35	3.48	5.64

(b) Within a short interval

Table 12 Search engine output variance across repeated queries. The left table shows results when two queries were issued one week apart, while the right table shows results when the two queries were issued within a short interval.

intersection).

- **pos_match:** The number of positions in the top-k where both lists contain the same item at the same rank.

As shown in Table 12, the search engine's returns are unstable. Even when calling the same query within a short interval, it still produces noticeably different results. The average overlap in the top-10 snippets is only about 7.67, with exact rank matches dropping to 5.64. When the same queries are reissued one week apart, the search engine's returns diverge more significantly. We show examples of inconsistencies in Figure 28.

F.3.2 Variances Introduced by Inference

Generating particularly long trajectories can also introduce variance. Small differences early in the process can lead to substantially divergent final responses. To observe the impact of this variability, we re-ran one short-form benchmark and two long-form benchmarks, comparing the outputs from two separate generations using GPT-4.1 with our auto-search pipeline.

As shown in Table 13, in 2Wiki, GPT-4.1 gives 29.3% difference final answers from 300 cases and obtains high variances in long-form tasks like Healthbench and ResearchQA as well.

Task	2Wiki	Healthbench	ResearchQA
Numbers	300	900	776
Diff	29.3	17.1	9.81

Table 13 Inference Variance. The Diff in 2Wiki refers to the difference in two final answers of the two trajectories under the same cases, while the Diff in Healthbench and ResearchQA represents the absolute difference in LLM judged scores.

F.3.3 Variances Introduced by Judge Models

When evaluating the same responses in different times, even if using the same model as a judge, inconsistent judgments may occur.

We use GPT-4.1 to evaluate the same trajectories twice and the results are shown in Table 14. We use GPT-4.1 to evaluate the same trajectories twice, and the results are shown in Table 14. The judgments show relatively high consistency and reliability on both short-form and long-form tasks.

	2Wiki	Healthbench	ResearchQA
1	67.67	37.67	66.18
2	66.33	37.51	66.43

Table 14 Judgement Variance.

Showcase of Search Engine's Returns

```
# Search Results I (show top-5 snippets)
Position: 1
Link: https://en.wikipedia.org/wiki/E._Howard_Hunt
Snippet: Everette Howard Hunt Jr. (October 9, 1918 - January 23, 2007) was an American
intelligence officer and author. From 1949 to 1970, Hunt served as an officer ...
Position: 2
Link: https://www.amazon.com/stores/author/B0034QAV74
Snippet: Top E. Howard Hunt titles · American Spy: My Secret History in the CIA, Watergate
and Beyond. American Spy: My Secret History in the CIA, Watergate and Beyond.
Position: 3
Link: https://www.fantasticfiction.com/h/e-howard-hunt/
Snippet: Everette Howard Hunt, Jr. was an American author and spy. He worked for the Central
Intelligence Agency (CIA) and later the White House under President ...
Position: 4
Link: https://www.goodreads.com/author/list/118536.E_Howard_Hunt
Snippet: E. Howard Hunt has 85 books on Goodreads with 2730 ratings. E. Howard Hunt's most
popular book is House Dick.
Position: 5
Snippet: E. Howard Hunt, a spy's spy. Hunt carried on writing spy novels long after the
Watergate scandal but the Peter Ward books are among his most popular series ...

# Search Results II
Position: 1
Link: https://en.wikipedia.org/wiki/E._Howard_Hunt
Snippet: Everette Howard Hunt Jr. (October 9, 1918 - January 23, 2007) was an American
intelligence officer and author. From 1949 to 1970, Hunt served as an officer ...
Position: 2
Link: https://www.goodreads.com/author/list/118536.E_Howard_Hunt
Snippet: E. Howard Hunt has 85 books on Goodreads with 2730 ratings. E. Howard Hunt's most
popular book is House Dick.
Position: 3
Link: https://www.amazon.com/E-Howard-Hunt/e/B0034QAV74/ref=dp_byline_cont_ebooks_1
Snippet: Follow E. Howard Hunt and explore their bibliography from Amazon's E. Howard Hunt
Author Page ... Howard Hunt. Most popular. American Spy: My Secret History ...
Position: 4
Link: https://www.fantasticfiction.com/h/e-howard-hunt/
Snippet: Everette Howard Hunt, Jr. was an American author and spy. He worked for the Central
Intelligence Agency (CIA) and later the White House under President ...
Position: 5
Link: https://spyscape.com/article/cia-spy-howard-hunt-confessions-of-a-watergate-plumber
Snippet: E. Howard Hunt, a spy's spy. Hunt carried on writing spy novels long after the
Watergate scandal but the Peter Ward books are among his most popular series ...
```

Figure 28 **Examples of Inconsistencies of Search Results.**

Figure 29 Metrics during RL training when starting from two different models (with and without SFT). "Pers. Rubric reward" refers to the reward over the search-based rubrics only (not including the rubrics generated as part of the RLER process). Starting from a model without cold start data ("no SFT") achieves higher train reward, but underperforms on downstream evaluations.

F.4 Mismatch between RL Training and Downstream Evaluation

During development of DR Tulu, we found that our RL training setup had some mismatch with our downstream performance: models that achieved the highest training reward did not necessarily achieve the highest downstream evaluation performance. To highlight this, compare the training reward of the "No SFT" and "Our SFT" models in Figure 29 against their performance in Figure 7. While starting directly from Qwen 3 ("No SFT") achieves highest train reward,it has dramatically lower downstream evaluation results (see Figure 7). It also displays significantly different behavior, using significantly more tool calls than the cold-started ("Our SFT") model.

This may be due to a few factors: first, our evaluations use rubrics generated in different manners to our own training rubrics (e.g., Healthbench uses expert-annotated rubrics), potentially leading to cases where test-time rubrics evaluate features not commonly tested in our training rubrics. Second, there may be reward hacking behavior during RL training, due to our use of an LM judge different to the judges used in downstream evaluation. Our in-loop judge model uses GPT-4.1-mini, while downstream evaluations use varied different models (e.g., SQAv2 uses Gemini Flash 2.5, DRB uses a mix of Gemini Pro 2.5 and Gemini Flash 2.5, and Healthbench uses GPT-4.1, all with varying prompts and evaluation harnesses). This may lead to our RL training optimizing for attributes preferred by GPT-4.1-mini, but not by downstream evaluation judges. We finally conjecture that another contributing factor may be the difference in model priors: different models may exploit rewards in different ways. For example, rollouts from a weaker model may contain fewer high-quality answers; when all answers are poor, the judge model may end up selecting based solely on spurious features rather than making meaningful quality comparisons. In contrast, when starting from the same initial model, we usually observe that reward improvements correlate well with downstream scores. We defer a deeper investigation of this mismatch phenomenon to future work, as addressing it would help improve the effectiveness of rubrics for RL training.

F.5 Qualitative Examples

We present one trajectory of our DR Tulu on the long-form task ResearchQA in Figures 34–36. The Response is truncated for brevity; see our full model responses examples at `https://dr-tulu.github.io/`.

Example Trajectory for OpenScholar SFT Data Part I

User

```
Please provide a detailed overview of theories and models related to Artificial Intelligence
(AI) in education. Include information on the foundational theories that underpin AI's role
in education, such as constructivism, behaviorism, and connectivism, and how these theories
inform AI-based learning tools. Additionally, explain prominent AI models and frameworks,
such as Intelligent Tutoring Systems (ITS), Adaptive Learning Systems, Learning Analytics,
and Natural Language Processing in education. Highlight their key features, applications, and
contributions to personalized learning, teacher efficiency, and student engagement. Please
reference recent scholarly works to support your response.
```

Assistant

```
<think>
Decompose the question: We need to (a) outline foundational learning theories (behaviorism,
constructivism, connectivism) and explicitly connect them to AI-in-education designs, and (b)
explain prominent AI models/frameworks (ITS, adaptive learning, learning analytics, NLP in
education), including features, applications, and contributions (personalization, teacher
efficiency, engagement), all supported by recent scholarly sources.
Assumptions: There are review/survey papers in AI in Education (AIED), Learning Analytics
(LAK), and Educational Data Mining (EDM) from 2020-2025 that synthesize how pedagogical
theories inform AI tools, and evaluations of ITS, adaptive systems, analytics, and NLP tools.
We should also pull specific examples and quantified impacts where possible (e.g., learning
gains from ITS, prediction performance in LA, efficacy of automated feedback).
Search plan:
1) Start with surveys/reviews linking AI in education to learning theories (2020-2025),
ideally in AIED, BJET, Computers \& Education, Review of Educational Research, LAK, EDM
venues.
2) Retrieve authoritative sources on ITS and adaptive learning (classic + recent advances
like knowledge tracing, deep KT, mastery learning, LLM tutoring).
3) Gather learning analytics frameworks and applications (early warning systems, dashboards,
ethics).
4) Collect NLP-in-education applications (automated scoring, feedback, dialogue tutoring,
LLMs) and evidence on teacher efficiency and student engagement.
First query will target a broad review tying theories and AI models together.
</think>
<call_tool name="snippet_search" year="2020-2025" limit="10"
fieldsOfStudy="Education,Computer Science">AI in education review behaviorism constructivism
connectivism intelligent tutoring systems adaptive learning learning analytics
NLP</call_tool>
```

Figure 30 **Example trajectory for OpenScholar SFT data Part I.**

Example Trajectory for OpenScholar SFT Data Part II

Functional Calling

```
<tool_output>
...
<snippet id="S_4yWBOUY">Title: The AI Revolution in Education: Will AI Replace or Assist
Teachers in Higher Education?
AI in education has evolved from computer-assisted instruction to intelligent tutoring
systems (ITS) and adaptive learning technologies, with key themes including personalized
feedback, natural language processing, and innovative tools like chatbots and virtual
reality, enhancing student engagement and facilitating knowledge co-creation between 2020 and
2025.</snippet>
<snippet id="S_KAymhlA">Title: AI Language Models as Educational Allies: Enhancing
Instructional Support in Higher Education
AI in education enhances instructional techniques, assessment methods, and personalized
learning through intelligent tutoring systems and adaptive learning, while also necessitating
further knowledge on technology to avoid misinformation, indicating a dynamic area of
research from 2020 to 2025.</snippet>
...
</tool_output>
```

Assistant

```
<think>
Reflection on first query results:
- The initial search returned broad reviews touching on AI branches (ML, DL, NLP) and
applications (ITS, adaptive learning, automated assessment) with some claims about
personalization and teacher support (S_c8C9s9m, S_NTanrK5, S_N8QWms7). These are relevant but
high-level and light on explicit connections to learning theories and quantitative evidence.
- S_tuhIPDv explicitly mentions ITS and NLP integrating behaviorism, constructivism, and
connectivism, which is directly pertinent, though I need stronger, peer-reviewed sources that
articulate these theory-to-tool mappings.
- For models/frameworks, I also need authoritative, recent evidence on ITS effectiveness,
adaptive learning via knowledge tracing, learning analytics frameworks (predictive models,
dashboards), and NLP applications (AWE, ASR, chatbots, LLMs) with impacts on personalization,
teacher efficiency, and engagement.

Gaps:
- Explicit theoretical grounding linking behaviorism, constructivism, and connectivism to AI
tools with scholarly backing.
- Quantitative or meta-analytic evidence for ITS/adaptive systems.
- Up-to-date learning analytics frameworks and outcomes.
- NLP in education evidence, including automated feedback and LLM-based tutoring.

Next query:
Search for papers that explicitly connect behaviorism, constructivism, and connectivism to
AI/ITS/adaptive learning frameworks in education, 2020-2025, in education and CS venues.
</think>
<call_tool name="snippet_search" year="2020-2025" limit="10"
fieldsOfStudy="Education,Computer Science">behaviorism constructivism connectivism artificial
intelligence in education theoretical framework</call_tool>
```

Functional Calling

```
...
```

Figure 31 **Example Trajectory for OpenScholar SFT Data Part II.**

Example Trajectory for OpenScholar SFT Data Part III

Assistant

```
<think>
We have initial broad reviews but need stronger, specific evidence tying learning theories to
AI designs and quantifying effects for ITS, adaptive learning, learning analytics, and NLP in
education. Key gaps: (1) explicit mapping of behaviorism/constructivism/connectivism to AI
paradigms with scholarly backing; (2) quantitative evidence for ITS/adaptive learning (e.g.,
learning gains, knowledge tracing); (3) learning analytics frameworks and teacher-facing
outcomes; and (4) NLP applications (AWE/ASR/chatbots/LLM tutors) with impacts on
personalization, efficiency, and engagement.

Next, I will search for meta-analyses or systematic reviews on ITS effectiveness and adaptive
learning from 2020-2025, then for learning analytics frameworks and NLP-in-education
evaluations.
</think>
<call_tool name="snippet_search" year="2020-2025" limit="10"
fieldsOfStudy="Education,Computer Science">intelligent tutoring systems meta-analysis
effectiveness learning gains 2020..2025 educational data mining knowledge tracing adaptive
learning</call_tool>
```

Functional Calling

```
...
```

Assistant

```
<think>
We have gathered broad reviews on AI in education and some mentions connecting pedagogical
theories to AI paradigms. We also have meta-analytic evidence for ITS effectiveness
summarized in recent reviews, but we still need: (a) clearer, recent sources on adaptive
learning mechanisms like knowledge tracing; (b) specific learning analytics frameworks and
teacher-facing applications; and (c) NLP-in-education evidence including automated feedback
and LLM-based tutoring impacts. We have performed four searches; we can do up to four more if
needed. However, current snippets include: a systematic review describing AI branches and
applications in education (S_c8C9s9m); papers situating AI paradigms within behaviorist,
constructivist, and connectivist frames (S_YRwbcNE, S_JCPVjqk, S_6KKZmn3); reviews
summarizing ITS/adaptivity and meta-analyses with effect sizes (S_lUcsvG5, S_sUEgQk4,
S_xkpGPkk); and reviews of NLP applications in language learning (S_KVwSGwG) and integrated
AI+learning analytics perspectives (S_RBEFb7h). While we could add more targeted evidence on
knowledge tracing and dashboards, the current evidence suffices to construct a theory-to-tool
mapping and to detail features/applications and contributions across ITS, adaptive learning,
learning analytics, and NLP with citations. I will structure the answer with sections: (1)
Foundations: behaviorism, constructivism, connectivism and their implications for AI tools;
(2) Intelligent Tutoring Systems; (3) Adaptive Learning Systems; (4) Learning Analytics; (5)
NLP in Education; (6) Contributions to personalized learning, teacher efficiency, and
engagement; (7) Emerging trends and considerations. I will ensure claims are supported by
retrieved snippets.
</think>
```

Figure 32 **Example trajectory for OpenScholar SFT data Part III.**

Example Trajectory for OpenScholar SFT Data Part IV

Assistant (Final Answer)

```
<answer>
# Overview

Artificial Intelligence (AI) in education encompasses techniques such as machine learning,
deep learning, and natural language processing that analyze learner data to personalize
instruction, generate feedback, and support pedagogical decision-making across intelligent
tutoring systems, adaptive platforms, and learning analytics tools <cite id="S_c8C9s9m">AI
has many branches and sub-branches, such as machine learning, deep learning, and natural
language processing; in education, AI supports intelligent tutoring systems, intelligent
agents, and intelligent collaborative learning systems, and supports teachers' decision
making by reporting real-time class statuses and personalized learning platforms</cite>.
Recent syntheses emphasize AI's rapid expansion into personalized feedback, chatbots, and
virtual environments for engagement and co-creation, while underscoring the need for
interdisciplinary integration with learning sciences <cite id="S_4yWB0UY">AI in education has
evolved from computer-assisted instruction to intelligent tutoring systems and adaptive
learning technologies, with key themes including personalized feedback, natural language
processing, and innovative tools like chatbots and virtual reality, enhancing student
engagement and facilitating knowledge co-creation</cite><cite id="S_RBEFb7h">The integration
of AI, particularly through learning analytics and NLP, is revolutionizing education by
enhancing personalized learning experiences and engagement, with advancements like large
language models improving performance in examinations and emphasizing collaboration between
AI developers and educators</cite>.

# Foundational Theories and How They Inform AI in Education
Behaviorism, cognitivism, constructivism, and connectivism offer complementary lenses for
designing AI-enabled learning experiences. Behaviorism emphasizes observable behavior change
through stimuli, repetition, and reinforcement, aligning with drill-and-practice, mastery
learning, and immediate feedback mechanisms common in early computer-assisted instruction and
some ITS components <cite id="S_6KKZmn3">The earliest learning theory, behaviorism, focuses
on human behaviors based on repetition and response to stimuli, enabling learning to be
observed and quantified</cite>. Constructivism positions learners as active constructors of
knowledge through authentic, collaborative, and reflective activity, motivating AI tools that
scaffold inquiry, support problem-based learning, and adaptively guide students through
complex tasks <cite id="S_6KKZmn3">Constructivism is learner-centered, emphasizing
collaborative, active, and reflective learning where students act autonomously and
self-directed</cite><cite id="S_XrZdksg">Constructivism encompasses situated cognition,
activity theory, experiential learning, anchored instruction, and authentic learning as
frameworks for educational technology, organized under a constructivist umbrella</cite>.
Connectivism reframes learning as networked, distributed across people and technologies,
shaping AI designs that augment learner agency through networked resources, recommendation,
and collaborative tools typical of MOOCs and social learning environments <cite
id="S_JCPVjqk">Connectivism, emerging as a response to technology's role, emphasizes learning
through external connections and technology performing cognitive tasks</cite><cite
id="S_TzKH9RC">Connectivism is grounded in interactions within networks, incorporating
aspects of behaviorism, cognitivism, and social constructivism, and has been influential as
an epistemological basis for MOOCs</cite>.
... (Response truncated for brevity; see our full dataset for complete prompts and examples.)
```

Figure 33 **Example trajectory for OpenScholar SFT data Part IV.**

Example Trajectory of DR Tulu on ResearchQA Part I

User

```
How do the narrow and broad approaches to CSR reporting differ in terms of their scope and
the stakeholders they address?
```

Assistant

```
<think>
Decompose the question: We need to define and differentiate the "narrow" versus "broad"
approaches to CSR reporting, specifically along two dimensions: (1) scope (what is reported)
and (2) stakeholders addressed (who benefits/engages with the report). Assumptions: This
terminology is used in CSR/ sustainability reporting frameworks and literature (e.g.,
AA1000AS, GRI, SASB, integrated reporting). Plan:
1) Search for authoritative definitions of "narrow" vs "broad" stakeholder engagement in CSR.
2) Find sources that explicitly define "narrow scope" CSR (e.g., compliance-only, triple
bottom line metrics) vs "broad scope" (stakeholder dialogue, materiality, governance,
long-term value).
3) Retrieve evidence on who the reports address (shareholders vs a wider stakeholder
community), including practitioner frameworks (e.g., AA1000AS stakeholder engagement
principle).
4) Cross-check with standards like GRI (comprehensive, wide stakeholder audience) and SASB
(focused, investor-material), to map scope vs audience.
First query: "narrow and broad approaches to CSR reporting definition stakeholders scope".
</think>
<call_tool name="google_search" num="5" gl="us" hl="en">narrow and broad approaches to CSR
reporting definition stakeholders scope</call_tool>
```

Functional Calling

```
<tool_output><snippet id=d7607bb2-0>
Title: Mandatory CSR and sustainability reporting: economic analysis and ...
URL: https://pmc.ncbi.nlm.nih.gov/articles/PMC8319709/
Snippet: Whether one chooses a narrow or a broad approach to CSR reporting depends on-among
other things-normative views about the intended scope and target audience.
</snippet>
<snippet id=d7607bb2-1>
Title: Stakeholders, stakeholder theory and Corporate Social ...
URL: https://jcsr.springeropen.com/articles/10.1186/s40991-024-00094-y
Snippet: This paper shows how stakeholder theory proactively moderates the strength of CSR in
social interactions, environmental protection, and sustainable development.
</snippet>
<snippet id=d7607bb2-2>
Title: Sustainability, corporate social responsibility, non-financial reporting ...
URL: https://www.sciencedirect.com/science/article/pii/S2352550922003116
Snippet: CSR is a corporate management approach that applies sustainability values in
business to promote social welfare within a company and outside it, employs ethical ...
</snippet>
...
```

Figure 34 **Example trajectory of DR Tulu on ResearchQA Part I.**

Example Trajectory of DR Tulu on ResearchQA Part II

Assistant

```
<think>Goal: Define and differentiate ""narrow"" vs ""broad"" approaches to CSR reporting by
(a) scope of what is reported and (b) stakeholders addressed. Strategy:
1) Find authoritative definitions and contrasts (preferably from academic papers and
reputable frameworks).
2) Use the first Google search results to identify key sources, then browse pages for precise
language on scope and audience.
3) Corroborate with recognized reporting frameworks (AA1000AS stakeholder engagement
principle; GRI vs SASB scope differences).
4) Ensure at least four searches: initial Google search done; next, targeted searches for
AA1000AS, GRI/SASB comparison, and a paper explicitly discussing narrow vs broad CSR
reporting scope.

Reflection on first search:
- PMC article (S_d7607bb2-0) mentions that approach choice depends on intended scope and
target audience-relevant framing.
- Sustainability Management Wiki (S_d7607bb2-3) claims narrow vs broad scope/target
group-potentially useful but I need more authoritative backing.
- Harvard Law corporate governance post (S_d7607bb2-5) discusses CSR report informing
stakeholders about addressing their concerns-useful for audience/stakeholder dialogue.
- NBER working paper (S_d7607bb2-6) gives an operational definition of narrowing scope to
securities-law-required info-useful for narrow scope exemplar.

What's missing: Explicit framework language (AA1000AS principle 1: stakeholder engagement as
a condition for materiality) and standard contrasts (GRI comprehensive vs SASB
issue-focused/investor-oriented). Next query: ""AA1000AS stakeholder engagement principle 1
materiality definition"".</think>
<call_tool name=""google_search"" num=""5"" gl=""us"" hl=""en"">AA1000AS stakeholder
engagement principle 1 materiality definition</call_tool>
```

Functional Calling

```
<tool_output><snippet id=59a6bf6a-0>
Title: Understanding the AA1000 Assurance Standard - LinkedIn
URL:
https://www.linkedin.com/pulse/understanding-aa1000-assurance-standard-enhancing-credibility-
vn1tf
Snippet: Materiality: Organizations must determine the relevance and significance of an issue
to stakeholders and the business, ensuring that all ...
</snippet>
<snippet id=59a6bf6a-1>
Title: AA1000 Stakeholder Engagement Standard - Accountability.org
URL: https://accountability.org/standards/aa1000-stakeholder-engagement
Snippet: The AA1000SES sets the global standard for conducting high-quality stakeholder
engagement by offering a simple, practical framework for empowering stakeholders.
</snippet>
<snippet id=59a6bf6a-2>
Title: [PDF] AA1000 STAKEHOLDER ENGAGEMENT STANDARD 2011
URL:
https://cdn2.hubspot.net/hubfs/2642721/Recursos/Guias\%20y\%20Estandares/AA1000/AA1000\%20Stake
holder\%20engagement\%20standard.pdf
Snippet: Inclusivity, materiality and responsiveness require a defined process of stakeholder
engagement that provides comprehensive and balanced involvement and ...
</snippet>
...
```

Figure 35 **Example trajectory of DR Tulu on ResearchQA Part II.**

Example Trajectory of DR Tulu on ResearchQA Part III

Assistant (Final Answer)

```
<think>
Plan: We need authoritative definitions and contrasts of ""narrow"" versus ""broad"" CSR
reporting, specifically along (1) reporting scope and (2) stakeholders addressed. I will:
- Use the initial Google results to identify key academic framing (scope and audience).
- Add AA1000AS to ground a ""broad"" approach via stakeholder engagement and materiality.
- Use GRI vs SASB comparisons to concretely illustrate broad (comprehensive,
multi-stakeholder) versus narrow (focused, investor-material) scope and audiences.
I have already run three searches and have relevant snippets: a peer-reviewed article noting
that approach choice depends on intended scope and target audience; examples of narrowing
scope to legally required information; and multiple GRI-SASB sources indicating GRI's broad
scope and audience versus SASB's focused/investor materiality. I will synthesize these,
noting where evidence is explicit (scope and audience differences) and where it is
inferential (mapping ""broad"" to AA1000AS stakeholder engagement/materiality). I will then
produce a structured answer with citations on each claim.

Remaining gaps: A direct paper explicitly defining ""narrow"" vs ""broad"" CSR reporting
terms would be ideal, but the PMC article and other sources suffice to explain the
distinction in scope and audience. I will proceed to synthesize.

Structure for answer:
- Definitions and framing: what ""narrow"" vs ""broad"" mean in scope and audience.
- Scope differences: compliance/required info vs comprehensive material issues.
- Stakeholders addressed: narrow (selected investors/regulators/creditors) vs broad (all
materially affected stakeholders including communities, customers, employees, suppliers,
NGOs).
- Standards mapping: AA1000AS (stakeholder engagement/materiality), GRI (broad), SASB
(narrow/investor materiality) as illustrative examples.

</think>
<answer>
# Short answer

- Scope: A narrow CSR reporting approach confines disclosure to a limited set of issues,
often legally required or compliance-oriented; a broad approach covers a wider set of
material issues identified through stakeholder engagement across economic, environmental, and
social dimensions. <cite id=""d7607bb2-6"">One way to narrow the scope of CSR reporting is to
confine it to information required by securities law.</cite> <cite id=""4a55f4d8-4"">GRI
requires identifying material topics to establish the scope and included issues covered by a
company's report.</cite> <cite id=""59a6bf6a-2"">AA1000's principles of inclusivity,
materiality, and responsiveness require a defined process of stakeholder engagement to
determine what to report.</cite>
- Stakeholders addressed: A narrow approach targets a limited audience (e.g., investors,
regulators, or specific creditors) and signals compliance; a broad approach addresses a wider
community of stakeholders (including employees, communities, customers, suppliers, NGOs)
through dialogue and reporting on issues that materially affect them. <cite
id=""d7607bb2-0"">Whether one chooses a narrow or broad approach depends on normative views
about the intended scope and target audience.</cite> <cite id=""d7607bb2-5"">CSR reports can
inform stakeholders about how a company has addressed their concerns, reflecting stakeholder
engagement.</cite> <cite id=""4a55f4d8-1"">GRI has a larger scope and broader audience, while
SASB enables deeper, sector-specific information for a more focused audience.</cite>

# What ""narrow"" CSR reporting entails
```

Figure 36 **Example trajectory of DR Tulu on ResearchQA Part III.**

Example Trajectory of DR Tulu on ResearchQA Part IV

Assistant (Final Answer)

```
- Reporting scope: Restricted to a small set of issues, typically where regulation or minimal
standards dictate disclosure; this constrains the breadth of topics and the depth of
information. <cite id=""d7607bb2-6"">Confining CSR reporting to information required under
securities laws is a concrete way to narrow scope.</cite>
- Stakeholders addressed: Primarily those with formal information rights or enforcement
powers (e.g., investors, regulators, creditors) rather than a broad stakeholder community;
the report often signals compliance rather than engaging a wider set of interest groups.
<cite id=""d7607bb2-0"">Choice of a narrow approach is linked to a narrower target
audience.</cite> <cite id=""d7607bb2-5"">A narrow CSR report addresses concerns raised by a
more limited set of stakeholders compared to a broad report.</cite>

# What ""broad"" CSR reporting entails

- Reporting scope: Comprehensive coverage of material issues determined through inclusive
stakeholder engagement and materiality assessments across environmental, social, and
governance domains; this typically includes balanced, context-sensitive issues with depth
where material. <cite id=""59a6bf6a-2"">Materiality and inclusivity require a defined
stakeholder engagement process to guide what is included.</cite> <cite
id=""4a55f4d8-4"">Materiality is used to establish the report's scope and included
issues.</cite> <cite id=""4a55f4d8-6"">GRI Standards are designed to be used by organizations
in any sector, implying broader coverage.</cite>
- Stakeholders addressed: A wide range of stakeholders who can affect or are affected by the
organization, including employees, communities, customers, suppliers, NGOs, and regulators,
with reporting intended to inform and facilitate dialogue. <cite id=""d7607bb2-0"">A broad
approach targets a broader audience.</cite> <cite id=""d7607bb2-5"">CSR reports in a broad
approach inform stakeholders about how their concerns have been addressed, reflecting
engagement.</cite> <cite id=""4a55f4d8-2"">GRI is designed to provide information to a broad
audience, while SASB dives deeper for a more specific one.</cite>

# How standards illustrate the difference

- AA1000AS (Accountability Principles): Embeds materiality in stakeholder engagement
(principles of inclusivity, materiality, and responsiveness), operationalizing a broad
approach that defines scope through dialogue with those affected and allocating limited
resources to material topics. <cite id=""59a6bf6a-2"">AA1000 emphasizes inclusivity,
materiality, and responsiveness via structured stakeholder engagement.</cite> <cite
id=""59a6bf6a-8"">A material topic is one that will substantively influence assessments,
decisions, actions, and performance of the organization and/or stakeholders.</cite>
- GRI: Broad, global framework that covers a wide range of sustainability topics and guides
scope via materiality assessments, addressing diverse stakeholder groups. <cite
id=""4a55f4d8-6"">GRI Standards are sector-applicable and designed for broad use, covering a
wide range of sustainability topics.</cite> <cite id=""4a55f4d8-4"">GRI requires identifying
material topics to establish scope.</cite> <cite id=""4a55f4d8-1"">GRI's scope is larger and
intended for a broader audience than SASB.</cite>
- SASB: Narrower, investor-oriented framework that focuses on industry-specific, financially
material issues, enabling deeper coverage for a focused audience (primarily investors). <cite
id=""4a55f4d8-1"">SASB enables deeper, sector-specific information for a more focused
audience than GRI.</cite> <cite id=""4a55f4d8-8"">SASB is focused on financially material ESG
issues in specific industries, i.e., a narrower scope and audience.</cite>
... (Response truncated for brevity; see our full example in our static demo.)
```

Figure 37 **Example trajectory of DR Tulu on ResearchQA Part IV.**