Deep Research as Rubric for Reinforcement Learning

Wangyi Mei[1,2], Zhouhong Gu[2], Zhenhan Bai[2,3], Yin Cai[1,2], Lefan Zhang[2], Zhenxin Ding[2], Bo Chen[2], Yan Gao[2], Yi Wu[2], Yao Hu[2]

[1]Fudan University, [2]Xiaohongshu Inc., [3]Beijing University of Posts and Telecommunications

{wymei24@m.,ycai25@m.}fudan.edu.cn,
zhenhanbai2023@bupt.edu.cn
{guzhouhong1,zhanglefan,dingzhenxin,chenbo}@xiaohongshu.com,
{wanjianyi,luyun2,xiahou}@xiaohongshu.com

Code: https://github.com/meiotoufa/DR-Rubric

Abstract

Open-ended reasoning and long-form generation tasks lack reliable automatic verification signals for reward-based policy optimization. Rubrics offer a promising alternative, but existing approaches treat them as given artifacts, either hand-crafted or prompt-generated, and often miss the task-specific, knowledge-intensive dimensions that matter most, distorting the reward signal. Our key observation is that *rubric construction is itself a research problem*: identifying what makes a response correct or insightful requires discovering and synthesizing external knowledge. We propose **Deep Research as Rubric (DR-rubric)**, a two-stage framework for constructing such rubrics. Stage I elicits domain facts, structural constraints, and failure modes through iterative multi-turn agentic search; Stage II distills this evidence into atomic, independently verifiable constraints for GRPO-based policy optimization. Because the model under training can serve as its own rubric generator, DR-rubric-8B supports *bootstrap rubric generation* without a frontier rubric generator. We evaluate on 6 benchmarks spanning agentic research and expert reasoning. Experiments show that DR-Rubric achieves strong competitive performance with only 1K–3K RL instances, where GPT-5-generated rubrics particularly benefit breadth coverage on agentic tasks, Gemini-generated rubrics yield the most balanced performance across agentic and expert reasoning tasks, and bootstrap rubrics exhibit a specialization-to-rebalancing evolution, reaching the strongest expert-reasoning scores at the third iteration. Results demonstrate that reframing rubric construction from static evaluation templates into an evidence-driven research process yields more scalable, fine-grained reward signals for open-ended tasks.

1 Introduction

Large language models (LLMs) have demonstrated remarkable capabilities in complex tasks such as open-ended reasoning and long-form generation. To truly unlock and harness these capabilities, reinforcement learning has become an indispensable component of the post-training stage. However, unlike tasks with clear ground truth such as mathematical proofs, open-ended problems lack definitive reference answers, making it difficult to produce automatic verification signals for policy optimization. *Rubrics*, structured evaluation criteria that decompose holistic quality judgments into fine-grained, interpretable dimensions, offer an effective pathway to address this absence of explicit rewards [1, 2, 3]. By rewarding each satisfied constraint separately, the feedback provides partial credit and gives reinforcement learning a dense and stable training signal.

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

Despite the strong potential of rubric-based approaches, existing construction methods typically treat rubrics as static, given artifacts, relying heavily on expensive manual authoring or simple prompt-based generation from large models [4]. While expert-authored rubrics carry high authority, this labor-intensive approach clearly lacks long-term scalability and maintainability in the face of massive, highly diverse training data distributions. Prompt-based methods infer rubrics from prompts or response data without explicit external evidence collection [5, 6]. We explicitly gather external evidence to identify knowledge-intensive, task-specific constraints. How to automatically generate such "one-to-one" customized evaluation criteria at scale constitutes the core technical challenge in this area.

Our key observation is that *rubric construction is itself a research problem*. Identifying what makes a response correct, comprehensive, or insightful for a given task requires discovering, gathering, and synthesizing relevant knowledge from external sources, which is the kind of structured inquiry that characterizes deep research. Based on this insight, we propose **Deep Research as Rubric (DR-Rubric)**, a systematic framework that reframes rubric construction as a two-stage, evidence-driven process. Deep research acquires external knowledge through iterative multi-turn retrieval and synthesis via agentic interaction [7, 8]. Incorporating this mechanism into the rubric generation process enables the model to actively collect, for each specific query, the objective facts, structural constraints, and common failure patterns needed to formulate evaluation criteria. In *Stage I: Information Elicitation*, an agentic research model conducts iterative multi-turn search to build a comprehensive evidence trace. In *Stage II: Rubric Synthesis*, the accumulated evidence is transformed into a formal rubric consisting of atomic, programmatically verifiable constraints that can be evaluated independently against candidate responses. Figure 1 illustrates the full pipeline.

To validate the effectiveness of DR-Rubric, we conduct extensive empirical analysis across 6 benchmark datasets spanning diverse research and reasoning tasks. Among 8B-scale models, DR-Rubric-8B with GPT-5-generated rubrics achieves strong agentic task performance, outperforming all 8B baselines on ResearchQA and DeepResearchBench, while Gemini-generated rubrics yield the most balanced profile, leading on LocalSearchBench and GPQA. More importantly, we explore and confirm the model's capacity for *bootstrap rubric generation*, the ability to self-generate customized rubrics. We find that the first bootstrap step produces strong reasoning performance (GPQA 57.0, MMLU 84.0), surpassing both benchmark-native and GPT-5-assisted rubrics on expert reasoning benchmarks, as self-generated rubrics closely match the model's capability boundary and induce stronger optimization pressure. However, this initial step also triggers capability specialization toward reasoning at the expense of agentic tasks. By Bootstrap Step 3, ResearchQA rises from 70.2 to 72.4 and the model reaches its best expert reasoning scores (MMLU-Pro 78.0, MMLU 85.3), surpassing GPT-5-assisted rubrics without an external rubric generator.

Our main contributions are as follows: **First**, we show why open-ended task evaluation requires customized rubrics and how to frame rubric construction as an evidence-grounded deep research process, linking knowledge synthesis to the design of evaluation criteria. **Second**, we propose DR-Rubric, a two-stage framework comprising Information Elicitation and Rubric Synthesis, that transforms task-specific external evidence into programmatically verifiable, fine-grained constraints. **Third**, we demonstrate the feasibility of bootstrap rubric generation and reveal a "specialization-to-rebalancing" dynamic across bootstrap iterations, where initial self-generated rubrics drive strong reasoning gains through capability specialization, and subsequent iterations progressively recover agentic task performance through structural rebalancing.

2 Related Work

Rubric-based rewards for open-ended post-training. For tasks without automatic verifiers, such as open-ended reasoning, long-form synthesis, and medical question answering, recent work converts rubrics into structured reward signals that guide post-training [9]. Decomposing judgments into interpretable criteria has been shown to yield more fine-grained and stable training signals than scalar or black-box preference rewards [1], and this idea has been extended to long-tail evaluation in LLM alignment [2]. Rubric dimensions can also serve as intermediate objectives in reinforcement learning, mitigating reward sparsity and guiding exploration in complex reasoning settings [3, 10]. Large-scale expert-authored rubrics for medical QA further illustrate both the power and the prohibitive cost of manual rubric construction [4]. A common thread is that these methods *assume the rubric itself is already available or can be directly obtained*: the research problem is how to use rubrics as reward,

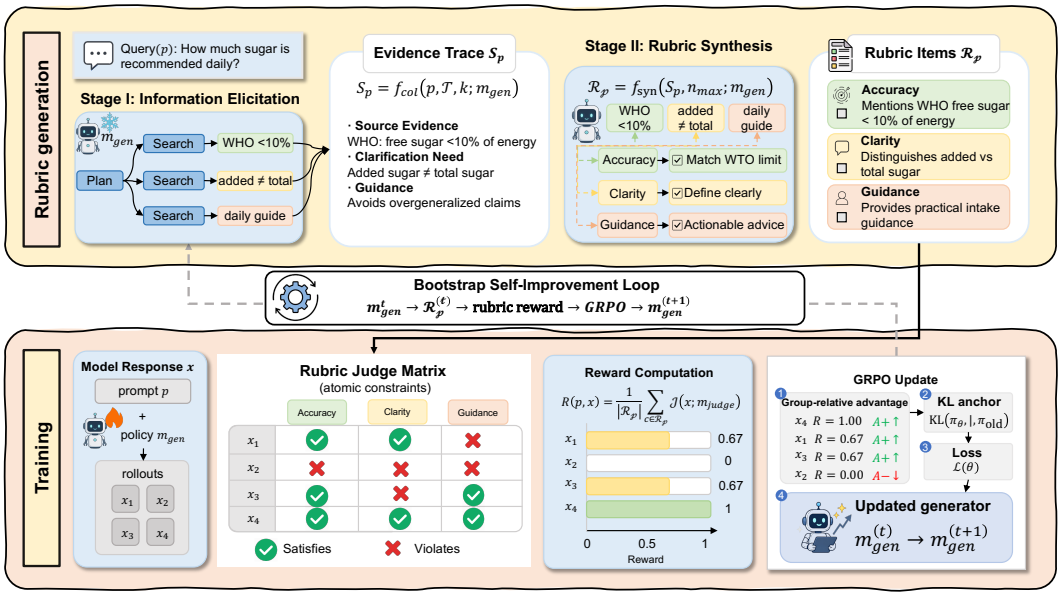


Figure 1: Overview of DR-Rubric. Given a task prompt p, evidence collection f_{col} conducts agentic research to build a structured evidence trace $\mathcal{S}_p$; constraint synthesis f_{syn} extracts atomic constraints to form the rubric $\mathcal{R}_p$. Rewards are computed per constraint and aggregated for GRPO training. The dashed arrow indicates *bootstrap rubric generation*: the trained policy $\pi_{\theta_{t+1}}$ replaces the external research model in subsequent iterations.

not how to construct them. Our work shifts attention from *rubric usage* to *rubric construction*, treating the design of task-specific evaluation criteria as a first-class learning problem.

Automated rubric generation with external evidence. To reduce reliance on manual specification, recent work automates rubric design. Some methods prompt LLMs to propose constraint-enhanced rewards for format, coherence, or rule compliance [11], synthesize criteria at scale via generation and filtering [5], or derive evaluation dimensions from pairwise comparisons [12]. These methods infer criteria from prompts or response comparisons without explicit external evidence collection. DR-Tulu [13] addresses this by grounding rubrics in retrieved context and evolving them during training by contrasting the policy's rollouts. However, a fixed external LM writes and revises the rubrics. DR-Rubric instead separates rubric construction into *information elicitation*, where the policy gathers explicit evidence through agentic research, and *rubric synthesis*, where it writes its own rubrics from that evidence, iterating across bootstrap rounds (see Appendix I for a detailed comparison with DR-Tulu's update mechanism).

Self-generated feedback and bootstrapped reward learning. Self-improvement methods demonstrate that models can provide feedback or supervision for their own training. Early work introduced self-critiques derived from human-written principles to iteratively refine responses [14]. More recent approaches further show that a model can act as both the generator and the evaluator, with self-generated feedback substituting for external supervision [15, 16]. However, the feedback in these approaches is typically scalar, preference-like, or principle-based: it tells the model *whether* a response is good but does not specify *what task-specific standards* it should satisfy. In contrast, DR-Rubric studies self-generated rubrics as structured, constraint-level reward specifications. We operationalize this capacity through bootstrap rubric generation: the trained policy constructs task-specific evaluation criteria for its own subsequent post-training. Complementary work optimizes parallel coordination [17] and recommendation policy gradients [18].

3 The DR-Rubric Framework

DR-Rubric transforms rubric construction from reliance on model-internal priors to a rigorous derivation grounded in external evidence. The framework comprises three sequential components: automated rubric generation (§3.1), reinforcement learning with rubrics (§3.2), and bootstrap rubric generation (§3.3). Table 1 defines all core symbols used throughout.

3.1 Rubric Generation

The rubric generation stage constructs a tailored rubric $\mathcal{R}_p$ for each query p, backed by external factual evidence. It consists of two steps: evidence collection and constraint synthesis.

Evidence Collection. Evidence collection takes as input a query p, an external tool set $\mathcal{T}$, and a maximum interaction budget k, and is driven by a generation model m_{gen} that conducts multi-round retrieval and verification. The output is an evidence trace $\mathcal{S}_p$ containing domain facts, structural constraints, and common failure modes:

$$\mathcal{S}_p = f_{col}(p, \mathcal{T}, k;\ m_{gen}). \tag{1}$$

The budget k caps the number of agent–environment interaction rounds, controlling the computational cost of the collection process.

Rather than assuming that evaluation criteria can be inferred directly from p, DR-Rubric treats missing task knowledge as a latent variable that must be explicitly acquired. The research process is iterative: at each step, the agent performs query formulation, evidence acquisition, and query refinement, where later queries are conditioned on earlier findings. This promotes coverage and reduces omission of critical constraints.

This formulation differs from standard retrieval-augmented generation in two key aspects. First, the objective is *requirement discovery* rather than answer production: exploration targets information that constrains evaluation rather than content that directly solves the task. Second, the iterative structure allows later queries to build on earlier findings, yielding an evidence trace $\mathcal{S}_p$ that serves as an explicit intermediate representation bridging task prompts and evaluation criteria.

Constraint Synthesis. An atomic constraint c is an indivisible single evaluation criterion. The rubric $\mathcal{R}_p$ is defined as the set of atomic constraints tailored to query p:

$$\mathcal{R}_p = \{c_1, c_2, \ldots, c_n\}. \tag{2}$$

Constraint synthesis distills the unstructured evidence trace $\mathcal{S}_p$ into this form, with n_{max} capping the rubric size to prevent redundancy and noise from overly large constraint sets:

$$\mathcal{R}_p = f_{syn}(\mathcal{S}_p, n_{max};\ m_{gen}). \tag{3}$$

Each constraint is phrased in natural language as either an affirmative requirement ("the response should cover at least two efficient attention methods") or a negation requirement ("the response should not claim that standard attention is $O(n \log n)$"). Both types reside in a single constraint set with no categorical distinction or differential weighting; they differ only in linguistic form. Negation also matters in retrieval [19, 20].

Crucially, constraints are not heuristically authored but are traceable to specific elements in $\mathcal{S}_p$. Together, evidence collection and constraint synthesis ensure that evaluation criteria derive from explicit external facts rather than parametric biases accumulated during pretraining.

3.2 Reinforcement Learning with Rubrics

The tailored rubric $\mathcal{R}_p$ is converted into a dense reward signal to drive parameter optimization of the policy model π_θ. This involves reward computation and policy update.

Reward Computation. The judge model m_{judge} independently scores each atomic constraint $c \in \mathcal{R}_p$ with a discrete judgment $J_c(x; m_{judge}) \in \{0, 1\}$, determining whether response x satisfies the constraint. In all main experiments, m_{judge} is DeepSeek-R1 prompted to judge one constraint at a time; it is separate from both the policy and the rubric generator. The final reward $R(p, x)$ is the

Table 1: Core notation for DR-Rubric.

Symbol	Definition	Description
p	Task Prompt	A raw query instance from the training set
m_{gen}	Generation Model	The model driving deep research and rubric synthesis
f_{col}	Evidence Collection	Multi-round deep-research evidence gathering process
$\mathcal{S}_p$	Evidence Trace	Set of facts and structural patterns collected for p
f_{syn}	Constraint Synthesis	Process converting evidence traces into normalized constraints
$\mathcal{R}_p$	Rubric	Full set of atomic constraints tailored to p
c	Atomic Constraint	An indivisible single evaluation criterion in the rubric
x	Response	Text generated by the model given p
m_{judge}	Judge Model	Model performing atomic constraint verification
$J_c(x; m_{judge})$	Judge Predicate	Discrete indicator of whether x satisfies constraint c
$R(p, x)$	Reward	Arithmetic mean of scores across all constraints
π_θ	Policy Model	The main model undergoing parameter optimization
$\pi_{\theta_{old}}$	Old Policy	Policy snapshot before the current update step
π_{ref}	Reference Model	Initial base model constraining update magnitude
G	Group Size	Number of responses sampled per prompt in GRPO
k	Max Steps	Upper limit on agent–environment interactions during research
n_{max}	Max Constraints	Maximum number of atomic constraints per rubric
ϵ	Clip Coefficient	Probability-ratio clipping width
β	KL Penalty	Regularization weight controlling policy divergence

arithmetic mean over all constraint scores, aggregating multi-dimensional quality assessment into a single reward signal:

$$R(p, x) = \frac{1}{|\mathcal{R}_p|} \sum_{c \in \mathcal{R}_p} J_c(x;\ m_{judge}). \tag{4}$$

This design distributes supervision across atomic evaluation criteria rather than assigning a single holistic score. Responses receive partial credit through selective constraint satisfaction, producing denser and more informative feedback than task-level binary rewards. Because each component corresponds to an interpretable requirement derived from $\mathcal{S}_p$, the reward remains grounded in evidence-backed task knowledge.

Policy Update. We use Group Relative Policy Optimization (GRPO), which estimates advantages by comparing rewards within a group of sampled responses. For each query p, we sample G responses $\{x_1, \ldots, x_G\}$ from $\pi_{\theta_{old}}$ and normalize their rubric rewards:

$$A_i = \frac{R(p, x_i) - \text{mean}(\{R(p, x_j)\}_{j=1}^G)}{\text{std}(\{R(p, x_j)\}_{j=1}^G) + \delta}. \tag{5}$$

Here $\delta > 0$ ensures numerical stability; equal-reward groups have zero advantages. The token-level probability ratio is

$$r_{i,j} = \frac{\pi_\theta(x_{i,j} \mid p, x_{i,<j})}{\pi_{\theta_{old}}(x_{i,j} \mid p, x_{i,<j})}. \tag{6}$$

GRPO maximizes a clipped surrogate objective with a KL penalty relative to the reference policy π_{ref}:

$$\mathcal{L}(\theta) = \frac{1}{G} \sum_{i=1}^{G} \frac{1}{|x_i|} \sum_{j=1}^{|x_i|} \left[\min(r_{i,j} A_i,\ \text{clip}(r_{i,j}, 1-\epsilon, 1+\epsilon) A_i) - \beta D_{i,j}\right], \tag{7}$$

where ϵ controls clipping, β weights regularization, and $D_{i,j}$ is the token-level KL penalty relative to π_{ref}.

3.3 Bootstrap Rubric Generation

The automated rubric generation stage relies on an external high-performance model m_{gen} to conduct deep research. In practice, however, continuously calling such closed-source models incurs significant cost and limits scalability. Bootstrap generation breaks this dependency: as the policy model π_θ

improves its reasoning capabilities, it gradually becomes capable of assuming the rubric generation role, so that rubric generation no longer requires an external model. The judge m_{judge} is kept fixed across iterations. To test whether self-improvement holds with no external model at all, Appendix E replaces m_{judge} with the initial model used for training (before any bootstrap update, kept fixed); the loop still improves, and the only external input is retrieved evidence.

In bootstrap iteration t, the current policy π_{θ_t} directly takes over the evidence collection and constraint synthesis responsibilities originally assigned to m_{gen}, generating tailored rubrics for new queries:

$$\mathcal{R}_{p,t} = f_{syn}(f_{col}(p, \mathcal{T}, k;\ \pi_{\theta_t}),\ n_{max};\ \pi_{\theta_t})\,. \tag{8}$$

The resulting rubrics then serve as the reward source driving the next round of policy update:

$$\pi_{\theta_{t+1}} = \text{GRPO}(\pi_{\theta_t},\ \mathcal{R}_{p,t}). \tag{9}$$

This recurrence forms a self-improvement loop bounded by the model's own capacity: the policy trained in one iteration writes the rubrics that supervise the next. We evaluate this loop empirically in Section 4, comparing single-step and multi-step bootstrap variants against the GPT-5- and Gemini-assisted baselines. The loop is not guaranteed to improve indefinitely; Section 4.4 characterizes when the bootstrap should stop.

4 Experiments

4.1 Setup

We evaluate on six benchmarks spanning agentic tasks (ResearchQA [21], DeepResearchBench [22], LocalSearchBench [23]) and expert reasoning (GPQA [24], MMLU-Pro [25], MMLU [26]). ResearchQA reports normalized five-level rubric coverage. All evaluations use a shared mock retrieval backend for controlled comparison (live-search results in Appendix J). Qwen3-8B-SFT is obtained by fine-tuning Qwen3-8B on 1K ReAct-format trajectories, each sampled once from Qwen3-32B-FP8 using the same task prompts as RL training and following an identical rollout protocol. All DR-Rubric-8B models are initialized from this Qwen3-8B-SFT model. This SFT stage is used to align the model with the multi-turn tool-interaction format and serves as the shared Stage-0 initialization; the subsequent DR-Rubric stage optimizes the model with GRPO on 1K instances per step using rubric-derived rewards. The training queries are disjoint from all six evaluation sets. For cumulative bootstrap variants, each step trains on the same 1K-instance set from the previous step's checkpoint, yielding 2K and 3K cumulative RL instances for BS-2 and BS-3 respectively. Rewards are judged by DeepSeek-R1. All models use training seed 42 and data-shuffle seed 1. Unless otherwise specified, we use group size G=4, lower and upper clipping widths of 0.2 and 0.28, KL coefficient β=0.001, learning rate 5×10^{-6}, and 5 epochs on 16×H800 GPUs. Full benchmark, data, and implementation details are in Appendices B, J, and M.

Baselines. We compare against Qwen2.5-7B, Qwen3-8B [27], DR-Tulu-SFT-8B and DR-Tulu-RL-8B [13], Search-R1-7B [28], and WebExplorer-8B [29]. Our variants use benchmark-native, GPT-5-generated [30], Gemini-generated [31], or bootstrap-generated rubrics from Steps 1–3.

4.2 Main Results

Training efficiency. Table 2 compares DR-Rubric-8B against SFT, web-agent RL, and outcome-based RL baselines. All DR-Rubric-8B models initialize from Qwen3-8B-SFT under an identical multi-turn tool-calling format; DR-Rubric-8B (GPT-5) and DR-Rubric-8B (Gemini) each use only 1K RL instances, while BS-3 uses three bootstrap steps of 1K each, yielding 3K cumulative RL instances. In contrast, prior baselines require substantially larger training budgets: DR-Tulu-SFT uses 16K instances, DR-Tulu-RL and WebExplorer each use 25K, and Search-R1 uses 90K. With 1K–3K RL instances, DR-Rubric-8B variants match or exceed these baselines on most benchmarks (Figure 2), showing that rubric rewards are highly data-efficient for RL (SFT and rubric-generation costs in Appendix B.2). Rubric-based RL also improves on the shared SFT initialization: BS-3 gains $+2.38$ points in the query-matched six-benchmark average (95% CI $[+1.12, +3.63]$), and all four rubric variants have intervals above zero (Appendix E).

Gains on agentic research tasks. DR-Rubric-8B (GPT-5) achieves 73.7 on ResearchQA and 43.0 on DeepResearchBench, exceeding the best baseline by +3.8 and +3.2 respectively (Table 2).

Table 2: Main results across 6 benchmarks spanning agentic tasks and expert reasoning. **Bold** = best within section; underline = second best within section. All our models are based on Qwen3-8B-SFT. Green/red superscripts on DR-Rubric-8B rows show the gap to the best baseline.

		Agentic Task			*Expert Reasoning*		
Method	**Training**	ResearchQA	DRBench	LocalSearch	GPQA	MMLU-Pro	MMLU
Qwen2.5-7B	—	65.5	30.7	**37.1**	52.3	**74.3**	79.3
Qwen3-8B	—	66.6	38.7	**37.1**	40.0	73.3	78.5
Qwen3-8B-SFT	SFT, 1K	**69.9**	39.4	36.0	50.8	73.8	81.8
Search-R1-7B	RL, 90K	63.1	33.6	34.2	39.0	68.0	78.0
DR-Tulu-SFT-8B	SFT, 16K	66.7	**39.8**	36.4	42.0	71.0	79.0
DR-Tulu-RL-8B	SFT+RL, 25K	67.1	35.5	36.5	44.0	69.0	79.0
WebExplorer-8B	SFT+RL, 25K	66.1	37.3	36.8	**56.0**	71.5	**83.8**
DR-Rubric-8B (Gemini)	**SFT+RL, 1K**	71.7 $^{+1.8}$	41.5 $^{+1.7}$	**40.0** $^{+2.9}$	**57.3** $^{+1.3}$	75.0 $^{+0.7}$	83.8 $^{0.0}$
DR-Rubric-8B (GPT-5)	**SFT+RL, 1K**	**73.7** $^{+3.8}$	**43.0** $^{+3.2}$	37.1 $^{0.0}$	54.0 $^{-2.0}$	74.8 $^{+0.5}$	80.3 $^{-3.5}$
DR-Rubric-8B (BS-3)	**SFT+RL, 3K**	72.4 $^{+2.5}$	39.5 $^{-0.3}$	36.4 $^{-0.7}$	55.8 $^{-0.2}$	**78.0** $^{+3.7}$	**85.3** $^{+1.5}$

DR-Rubric-8B (Gemini) further excels on LocalSearchBench (40.0, +2.9 over the best baseline), indicating that Gemini-generated rubrics provide complementary coverage for local search tasks. These benchmarks reward covering latent, multi-dimensional criteria, which is the level at which atomic rubric constraints supply training signal.

Reasoning transfer. DR-Rubric-8B (BS-3) leads on MMLU-Pro (78.0, +3.7) and MMLU (85.3, +1.5), while DR-Rubric-8B (Gemini) achieves the highest GPQA score (57.3, +1.3 over WebExplorer-8B). Rubric-based RL thus transfers beyond agentic tasks to expert reasoning benchmarks across all rubric sources.

Divergent capability profiles. The three DR-Rubric-8B variants show complementary strengths: GPT-5 rubrics expand reasoning *breadth* (criteria coverage on agentic tasks); Gemini rubrics achieve the best balanced performance across both agentic and expert reasoning tasks; bootstrap rubrics deepen reasoning *depth* (general reasoning benchmarks). The next section separates matched source comparisons from bootstrap evolution.

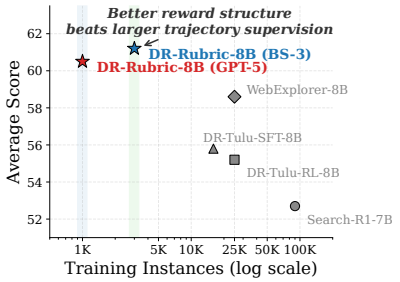


Figure 2: Training efficiency: per-benchmark scores vs. training instances (log scale). DR-Rubric-8B achieves the strongest results with significantly fewer training instances than all baselines.

Isolating the gains. Three controls separate the rubric reward from other factors (Appendix E). (i) With the same SFT initialization, data, and GRPO budget, replacing rubric rewards by a holistic DeepSeek-R1 score lowers the macro-average from 60.77 (BS-1) to 58.08, with rubric rewards better on all six benchmarks. (ii) Without SFT, bootstrap from Qwen3-8B still improves the average from 55.70 to 57.25. (iii) When the frozen Qwen3-8B-SFT checkpoint serves as the judge, so that no external model takes part after initialization, the average rises from 58.62 to 60.28 at BS-2.

Scaling beyond 8B. On Qwen3-14B and Qwen3-30B-A3B, bootstrap variants exceed scale-matched baselines, including DeepSeek-R1-Distill-Qwen-14B [32], WebThinker-R1-14B [33], and Tongyi-DeepResearch-30B-A3B [34], on ResearchQA and LocalSearchBench at every step, with mixed results on DRBench (Appendix C). Fresh-start results across all three scales appear in Appendix D.

4.3 Rubric Sources and Bootstrap Evolution

Table 3 supports two separate comparisons. GPT-5, Gemini, and BS-1 share Qwen3-8B-SFT, the DeerFlow elicitation harness, the training set, a one-step GRPO budget, atomic rewards, and evaluation; only the rubric-generating model differs, so these rows form a harness-matched source comparison. BS-1→BS-3 traces the full bootstrap process (Section 3.3), where the rubric generator,

Table 3: Rubric sources and bootstrap evolution. All models use Qwen3-8B-SFT + GRPO with atomic rubric reward. Elicit. = rubric elicitation harness; GPT-5, Gemini, and BS-1 form the harness-matched comparison. Bench. Rubric = benchmark-native evaluation criteria; GPT-5 = GPT-5-generated rubrics; Gemini = Gemini-generated rubrics; BS-1/2/3 = bootstrap steps. DRBench = DeepResearchBench; LocalSearch = LocalSearchBench. **Bold** = best; underline = second best.

		Agentic Task			*Expert Reasoning*		
Method	**Elicit.**	ResearchQA	DRBench	LocalSearch	GPQA	MMLU-Pro	MMLU
Bench. Rubric	n/a	73.2	42.6	37.1	54.3	71.5	81.0
DR-Rubric-8B (Gemini)	DeerFlow	71.7	41.5	**40.0**	**57.3**	75.0	83.8
DR-Rubric-8B (GPT-5)	DeerFlow	**73.7**	**43.0**	37.1	54.0	74.8	80.3
DR-Rubric-8B (BS-1)	DeerFlow	70.2	39.2	36.7	57.0	77.5	84.0
DR-Rubric-8B (BS-2)	ReAct	72.4	38.3	36.2	56.5	74.3	83.5
DR-Rubric-8B (BS-3)	ReAct	72.4	39.5	36.4	55.8	**78.0**	**85.3**

the policy initialization, and (for BS-2/BS-3) the policy-native ReAct interface change together. Benchmark-native rubrics are not generated through DeerFlow.

Benchmark-native rubrics as an in-distribution reference. Bench. Rubric is built from each training set's own evaluation criteria, so it matches the style of the test-time scoring criteria. It achieves strong agentic scores (ResearchQA 73.2, DRBench 42.6, tied best on LocalSearch at 37.1), yet GPT-5, Gemini, and bootstrap variants all surpass it on several benchmarks.

GPT-5 rubrics: coverage-oriented. GPT-5 rubrics achieve the highest scores on ResearchQA and DRBench (Table 3), reflecting the generator's capacity to produce richer evaluation dimensions, while the best MMLU and MMLU-Pro scores come from BS-3. GPT-5 rubrics thus act as a strong *coverage reward*.

Gemini rubrics: balanced across task types. Gemini rubrics achieve the best LocalSearch score (40.0) and the highest GPQA (57.3), demonstrating strong performance on both agentic and expert reasoning tasks. Compared to GPT-5, Gemini rubrics trade some agentic coverage for better reasoning transfer, yielding the most balanced profile across all six benchmarks.

Bootstrap: specialization-to-rebalancing. Bootstrap rubrics exhibit a non-monotonic trajectory (Figure 3): BS-1 achieves strong reasoning scores (GPQA 57.0, MMLU 84.0); BS-2 sees reasoning scores dip; BS-3 recovers and achieves the best expert reasoning results (MMLU-Pro 78.0, MMLU 85.3). From BS-1 to BS-3 the macro-average changes by $+0.4$, MMLU-Pro and MMLU rise, and GPQA decreases at each step. Practical guidance follows from the matched comparison: GPT-5 rubrics for coverage-heavy agentic tasks, Gemini rubrics for balanced performance, and bootstrap rubrics for reasoning-oriented settings where no external rubric generator is available.

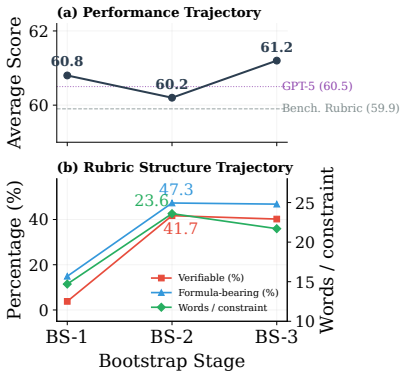


Figure 3: Bootstrap trajectory. (a) Performance is non-monotonic. (b) Verifiability and formula usage surge at BS-2 then stabilize, while constraint verbosity peaks then retracts.

Fresh-start bootstrap. Fresh-start bootstrap resets each round to a fixed initialization and uses the trained policy to generate the next round's rubrics. At 8B, BS-3 matches cumulative BS-3 within 0.4 points on five benchmarks and 1.5 on GPQA. Appendix D reports the same procedure at 8B, 14B, and 30B-A3B.

Effect of rubric granularity. Beyond rubric source, rubric *granularity* also shapes the reward signal. We vary the rubric generation parameters (n_{dim}, n_{cons}) to control rubric granularity, training four DR-Rubric-8B (BS-1) variants that differ only in rubric size. Table 4 reveals a clear task-dependent pattern: agentic search tasks score highest with the broadest rubrics (X-Large 50.6 vs.

Table 4: Rubric granularity has task-dependent effects. All variants are DR-Rubric-8B (BS-1) and differ only in rubric size. Dim./Cons. = dimension / constraint range used during rubric generation; Agentic and Expert = mean over the three benchmarks in each group. **Bold** = best per column.

		Agentic Task				*Expert Reasoning*			
Rubric Size	**Dim./Cons.**	ResearchQA	DRBench	LocalSearch	Agentic	GPQA	MMLU-Pro	MMLU	Expert
Compact	2–4 / 5–8	70.2	39.2	36.7	48.7	**57.0**	**77.5**	**84.0**	**72.8**
Medium	7–10 / 15–25	62.2	37.3	36.7	45.4	52.0	74.0	80.8	68.9
Large	10–14 / 25–30	70.3	**41.7**	36.3	49.4	55.5	71.3	83.0	69.9
X-Large	15–20 / 40–50	**71.9**	41.6	**38.3**	**50.6**	54.0	73.3	79.8	69.0

Compact 48.7, +1.9), while expert reasoning tasks peak under compact rubrics (72.8 vs. X-Large 69.0, −3.8).

4.4 Bootstrap Stability Limit

In our experiments, bootstrap performance peaks within two to three iterations, motivating an empirical stopping criterion. Final models stop at BS-2 or BS-3; to study what happens beyond this range, we deliberately continue bootstrapping on Qwen3-30B-A3B through BS-5 as a stress test, with Qwen3-14B as a second scale. The question is whether degradation can be detected before task performance collapses.

Polarization as an early-warning signal. Reward polarization, the fraction of rewards at 0 or $\geq$0.99, rises before downstream collapse (Figure 4). On 30B-A3B it stays low at BS-1 and BS-2 (0.096, 0.054) and exceeds 0.15 at BS-3 (0.195); agentic scores remain at 51.1 at BS-3 but fall to 22.7 at BS-4, and BS-5 fails with vanishing gradients (Appendix L). Internal reward meanwhile rises from 0.24 to 0.66, consistent with reward over-optimization. Adjacent-checkpoint KL does not expose this divergence, because each step measures drift only from the preceding checkpoint, and aggregate rubric statistics stay stable from BS-3 to BS-4 (Appendix E). These observations motivate a polarization-based stopping rule:

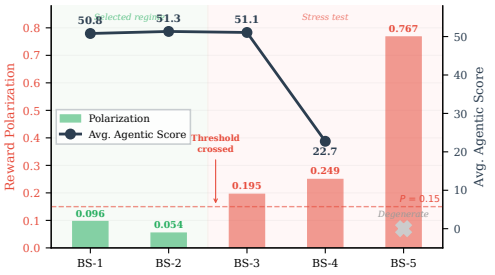


Figure 4: Polarization rises before downstream collapse (30B-A3B). Polarization (bars, left axis) exceeds the absolute cutoff of 0.15 (dashed line) at BS-3; agentic score (line, right axis) collapses at BS-4. Extended training analysis in Appendix L.

$$\tau_n = \min\big(0.15,\ 2\,P_{n-1}\big), \qquad \text{stop if } P_n > \tau_n, \tag{10}$$

where P_n denotes the reward polarization at bootstrap step n. The rule stops bootstrapping when polarization exceeds 0.15 or more than doubles from the preceding step. It flags BS-3 in both the 14B and 30B-A3B runs (Appendix L).

4.5 Intrinsic Rubric Quality

Different rubric sources produce structurally different rubrics, which helps interpret the non-monotonic bootstrap trajectory. Table 5 reports intrinsic metrics on 998 nonempty GPT-5 rubrics and 999 per bootstrap step from the same query set. We operationalize *dimensions* as Markdown #### headers and *constraints* as bullet-list items within each dimension, excluding GPT-5 assessment goals and nested scoring tiers (see Appendix F for full definitions).

GPT-5 rubrics: broad, terse, highly verifiable. GPT-5 rubrics average 6.50 dimensions and 24.39 constraints per rubric, far exceeding benchmark-native rubrics (2.81 dimensions, 8.90 constraints) and all bootstrap variants. Each constraint is short (4.8 words) but nearly always verifiable (96.4%) and formula-bearing (98.8%). This many-axis, fine-grained structure is consistent with GPT-5's coverage advantage: more dimensions yield more independent scoring axes, and terse verifiable constraints enable precise per-criterion credit assignment.

Table 5: Intrinsic rubric quality across rubric sources (998 nonempty GPT-5 rubrics, 999 per bootstrap step from the same query set; 899 for Bench. Rubric). Verifiable (%) = fraction of rubrics containing numerical thresholds (detected via regex for $\geq/\leq/>/<$ followed by digits). Jaccard = word-level similarity across steps (100-pair sample). $\pm$ values denote standard deviation across rubrics; for binary indicators (%), $\pm$ values denote standard error. †GPT-5 rubrics. ‡Benchmark-native rubrics aggregated from 4 benchmarks with expert-authored evaluation criteria (HealthBench, ResearchQA, RARMedicine, RARScience). ResearchQA rubrics are flat question lists without explicit dimension grouping (counted as 1 dimension); the three benchmarks with dimension structure average 3.71 dimensions.

Metric	**Bench.**‡	**GPT-5**†	**Step 1**	**Step 2**	**Step 3**
Dimensions / rubric	2.81 ± 1.38	6.50 ± 1.12	4.68 ± 1.38	3.59 ± 0.66	3.61 ± 0.55
Constraints / rubric	8.90 ± 3.30	24.39 ± 14.76	10.49 ± 2.90	10.22 ± 2.40	10.04 ± 2.05
Words / constraint	26.7 ± 40.0	4.8 ± 4.6	14.7 ± 2.1	23.6 ± 10.0	21.7 ± 8.9
Verifiable (%)	2.1 ± 0.5	96.4 ± 0.6	3.8 ± 0.6	41.7 ± 1.6	40.2 ± 1.6
With formulas (%)	16.1 ± 1.2	98.8 ± 0.3	14.9 ± 1.1	47.3 ± 1.6	46.8 ± 1.6
Type-token ratio	$0.525 \pm .083$	$0.320 \pm .055$	$0.515 \pm .055$	$0.520 \pm .067$	$0.529 \pm .059$
Cross-step Jaccard	—	0.032–0.034 (GPT-5 vs. BS);		0.130–0.148 (BS all pairs)	

Bootstrap evolves a different reward style. Bootstrap rubrics do not converge toward GPT-5 but evolve in the opposite direction (Figure 3): dimensions *decrease* ($4.68 \rightarrow 3.59 \rightarrow 3.61$) while constraint length *increases* ($14.7 \rightarrow 23.6 \rightarrow 21.7$ words), and verifiability jumps from Step 1 to Step 2 then saturates ($3.8\% \rightarrow 41.7\% \rightarrow 40.2\%$). This structural shift co-varies with the performance pattern in Table 3: **BS-1** stays close to benchmark-native style (broad, with fewer explicit numerical thresholds), providing strong reasoning supervision that peaks on GPQA and MMLU. **BS-2** introduces more threshold-based constraints, coinciding with lower reasoning scores. **BS-3** consolidates: constraint length decreases slightly (21.7 words), vocabulary diversity rises (TTR 0.529 vs. 0.520), recovering performance across benchmarks. Low cross-step Jaccard reflects lexical turnover; in a query-matched case (Table 19, Appendix H), it comes from a shift from naming hazards (BS-1) to specifying evidence such as particle-size ranges (BS-2) to explaining mechanisms (BS-3).

Rubric style and inference behavior. Two ResearchQA cases where only the training rubrics differ (Appendix G) suggest that rubric structure shapes tool use. On a question about the prevalence, clinical features, and treatment outcomes of childhood tuberculosis in Nigeria, the GPT-5-trained model searches all three facets together, while the Benchmark-trained model searches only prevalence. With identical search queries, GPT-5's terse constraints (4.8 words) lead to extracting discrete facts such as drug names and FEV1 cutoffs that the Benchmark-trained response misses (0.875 vs. 0.500).

5 Conclusion

We presented DR-Rubric, a framework that reframes rubric construction as a deep research problem grounded in externally acquired evidence. Experiments on 6 benchmarks show that evidence-grounded rubrics yield competitive 8B-scale performance with only 1K–3K RL instances, and that bootstrap training, where the model writes its own rubrics, matches or exceeds benchmark-native rubrics without an external rubric generator. Under a matched harness, rubric sources specialize: GPT-5 rubrics favor coverage on agentic tasks, Gemini rubrics are the most balanced, and bootstrap rubrics give the strongest expert-reasoning transfer.

Limitations. Performance depends on retrieval quality: for the fresh-start BS-3 checkpoint, switching to live search reduces scores from 72.6 to 65.8 on ResearchQA and from 39.9 to 31.7 on DRBench (Appendix J). Bootstrap gains vary across tasks, and further iterations can increase rubric rewards while reducing downstream performance.

Acknowledgments and Disclosure of Funding

This work was supported by Xiaohongshu Inc. The authors declare no additional competing interests beyond the affiliations listed above.

References

[1] Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Bing Liu, and Sean Hendryx. Rubrics as rewards: Reinforcement learning beyond verifiable domains. *CoRR*, abs/2507.17746, 2025.

[2] Junkai Zhang, Zihao Wang, Lin Gui, Swarnashree Mysore Sathyendra, Jaehwan Jeong, Victor Veitch, Wei Wang, Yunzhong He, Bing Liu, and Lifeng Jin. Chasing the tail: Effective rubric-based reward modeling for large language model post-training. *CoRR*, abs/2509.21500, 2025.

[3] Yang Zhou, Sunzhu Li, Shunyu Liu, Wenkai Fang, Jiale Zhao, Jingwen Yang, Jianwei Lv, Kongcheng Zhang, Yihe Zhou, Hengtong Lu, Wei Chen, Yan Xie, and Mingli Song. Breaking the exploration bottleneck: Rubric-scaffolded reinforcement learning for general LLM reasoning. *CoRR*, abs/2508.16949, 2025.

[4] Rahul K. Arora, Jason Wei, Rebecca Soskin Hicks, Preston Bowman, Joaquin Quiñonero Candela, Foivos Tsimpourlas, Michael Sharman, Meghan Shah, Andrea Vallone, Alex Beutel, Johannes Heidecke, and Karan Singhal. Healthbench: Evaluating large language models towards improved human health. *CoRR*, abs/2505.08775, 2025.

[5] Tianci Liu, Ran Xu, Tony Yu, Ilgee Hong, Carl Yang, Tuo Zhao, and Haoyu Wang. Openrubrics: Towards scalable synthetic rubric generation for reward modeling and LLM alignment. *CoRR*, abs/2510.07743, 2025.

[6] Lipeng Xie, Sen Huang, Zhuo Zhang, Anni Zou, Yunpeng Zhai, Dingchao Ren, Kezun Zhang, Haoyuan Hu, Boyin Liu, Haoran Chen, Zhaoyang Liu, and Bolin Ding. Auto-rubric: Learning to extract generalizable criteria for reward modeling. *CoRR*, abs/2510.17314, 2025.

[7] Ganlin Xu, Linghao Zhang, Zhitao Yin, Hongda Xi, Chen Yang, Jiaqing Liang, Weijia Lu, Sihang Jiang, Yanghua Xiao, and Deqing Yang. When RAG meets query planning: Logical query trees for resolving exploratory reasoning problems. *arXiv preprint arXiv:2607.00508*, 2026.

[8] Wangyi Mei, Ganlin Xu, Zhoujia Zhang, Jiaqing Liang, Weijia Lu, Jinhui Huang, Chen Yang, Xiaodong Zhang, Zhifei Yang, Xiaofeng Ma, and Deqing Yang. Learning to think in blocks: A prior-guided reinforcement learning framework for deep research. In *Findings of the Association for Computational Linguistics: EMNLP 2026*, pages 22566–22584. Association for Computational Linguistics, 2026.

[9] Jacob Dineen, Aswin RRV, Qin Liu, Zhikun Xu, Xiao Ye, Ming Shen, Zhaonan Li, Shijie Lu, Chitta Baral, Muhao Chen, and Ben Zhou. QA-LIGN: aligning llms through constitutionally decomposed QA. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng, editors, *Findings of the Association for Computational Linguistics: EMNLP 2025, Suzhou, China, November 4-9, 2025*, pages 20619–20642. Association for Computational Linguistics, 2025.

[10] Zenan Huang, Yihong Zhuang, Guoshan Lu, Zeyu Qin, Haokai Xu, Tianyu Zhao, Ru Peng, Jiaqi Hu, Zhanming Shen, Xiaomeng Hu, Xijun Gu, Peiyi Tu, Jiaxin Liu, Wenyu Chen, Yuzhuo Fu, Zhiting Fan, Yanmei Gu, Yuanyuan Wang, Zhengkai Yang, Jianguo Li, and Junbo Zhao. Reinforcement learning with rubric anchors. *CoRR*, abs/2508.12790, 2025.

[11] Jianghao Chen, Wei Sun, Qixiang Yin, Lingxing Kong, Zhixing Tan, and Jiajun Zhang. ACE-RL: adaptive constraint-enhanced reward for long-form generation reinforcement learning. *CoRR*, abs/2509.04903, 2025.

[12] MohammadHossein Rezaei, Robert Vacareanu, Zihao Wang, Clinton Wang, Bing Liu, Yunzhong He, and Afra Feyza Akyürek. Online rubrics elicitation from pairwise comparisons. *CoRR*, abs/2510.07284, 2025.

[13] Rulin Shao, Akari Asai, Shannon Zejiang Shen, Hamish Ivison, Varsha Kishore, Jingming Zhuo, Xinran Zhao, Molly Park, Samuel G. Finlayson, David A. Sontag, Tyler Murray, Sewon Min, Pradeep Dasigi, Luca Soldaini, Faeze Brahman, Wen-tau Yih, Tongshuang Wu, Luke Zettlemoyer, Yoon Kim, Hannaneh Hajishirzi, and Pang Wei Koh. DR tulu: Reinforcement learning with evolving rubrics for deep research. *CoRR*, abs/2511.19399, 2025.

[14] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, Jamie Kerr, Jared Mueller, Jeffrey Ladish, Joshua Landau, Kamal Ndousse, Kamile Lukosuite, Liane Lovitt, Michael Sellitto, Nelson Elhage, Nicholas Schiefer, Noemi Mercado, Nova DasSarma, Robert Lasenby, Robin Larson, Sam Ringer, Scott Johnston, Shauna Kravec, Sheer El Showk, Stanislav Fort, Tamera Lanham, Timothy Telleen-Lawton, Tom Conerly, Tom Henighan, Tristan Hume, Samuel R. Bowman, Zac Hatfield-Dodds, Ben Mann, Dario Amodei, Nicholas Joseph, Sam McCandlish, Tom Brown, and Jared Kaplan. Constitutional AI: harmlessness from AI feedback. *CoRR*, abs/2212.08073, 2022.

[15] Zixiang Chen, Yihe Deng, Huizhuo Yuan, Kaixuan Ji, and Quanquan Gu. Self-play fine-tuning converts weak language models to strong language models. *CoRR*, abs/2401.01335, 2024.

[16] Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. Self-rewarding language models. *CoRR*, abs/2401.10020, 2024.

[17] Yexiong Lin, Shanshan Ye, Yu Yao, Zhen Fang, Bo Han, and Tongliang Liu. SquidAgent: Parallelize wisely, coordinate efficiently. *arXiv preprint arXiv:2610.08647*, 2026.

[18] Hongru Hou, Tiehua Mei, Denghui Geng, Jinhui Huang, Ao Xu, Hengrui Chen, Jiaqing Liang, and Deqing Yang. ProRL: Effective reinforcement learning for proactive recommendation via rectified policy gradient estimation. *arXiv preprint arXiv:2605.28293*, 2026.

[19] Ganlin Xu, Zhitao Yin, Linghao Zhang, Jiaqing Liang, Weijia Lu, Xiaodong Zhang, Zhifei Yang, Sihang Jiang, and Deqing Yang. ComLQ: Benchmarking complex logical queries in information retrieval. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pages 34115–34123, 2026.

[20] Ganlin Xu, Zhoujia Zhang, Wangyi Mei, Jiaqing Liang, Weijia Lu, Xiaodong Zhang, Zhifei Yang, Xiaofeng Ma, Yanghua Xiao, and Deqing Yang. Logical consistency is vital: Neural-symbolic information retrieval for negative-constraint queries. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 1828–1847. Association for Computational Linguistics, 2025.

[21] 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. *CoRR*, abs/2509.00496, 2025.

[22] Mingxuan Du, Benfeng Xu, Chiwei Zhu, Xiaorui Wang, and Zhendong Mao. Deepresearch bench: A comprehensive benchmark for deep research agents. *CoRR*, abs/2506.11763, 2025.

[23] Hang He, Chuhuai Yue, Chengqi Dong, Mingxue Tian, Zhenfeng Liu, Jiajun Chai, Xiaohan Wang, Yufei Zhang, Qun Liao, Guojun Yin, Wei Lin, Chengcheng Wan, Haiying Sun, and Ting Su. Localsearchbench: Benchmarking agentic search in real-world local life services. *CoRR*, abs/2512.07436, 2025.

[24] David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof q&a benchmark. *CoRR*, abs/2311.12022, 2023.

[25] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. In Amir Globersons, Lester Mackey, Danielle Belgrave, Angela Fan, Ulrich Paquet, Jakub M. Tomczak, and Cheng Zhang, editors, *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information*

Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024, 2024.

[26] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net, 2021.

[27] Qwen Team. Qwen3 technical report. *CoRR*, abs/2505.09388, 2025.

[28] 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. *CoRR*, abs/2503.09516, 2025.

[29] Junteng Liu, Yunji Li, Chi Zhang, Jingyang Li, Aili Chen, Ke Ji, Weiyu Cheng, Zijia Wu, Chengyu Du, Qidi Xu, Jiayuan Song, Zhengmao Zhu, Wenhu Chen, Pengyu Zhao, and Junxian He. WebExplorer: Explore and Evolve for Training Long-Horizon Web Agents. *CoRR*, abs/2509.06501, 2025.

[30] OpenAI. Openai GPT-5 system card. *CoRR*, abs/2601.03267, 2026.

[31] Google DeepMind. Gemini 3.1 pro model card. `https://deepmind.google/models/model-cards/gemini-3-1-pro/`, February 2026. Accessed: 2026-05-30.

[32] DeepSeek-AI. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *CoRR*, abs/2501.12948, 2025.

[33] Xiaoxi Li, Jiajie Jin, Guanting Dong, Hongjin Qian, Yongkang Wu, Ji-Rong Wen, Yutao Zhu, and Zhicheng Dou. Webthinker: Empowering large reasoning models with deep research capability. *CoRR*, abs/2504.21776, 2025.

[34] Baixuan Li, Bo Zhang, Dingchu Zhang, Fei Huang, Guangyu Li, Guoxin Chen, Huifeng Yin, Jialong Wu, Jingren Zhou, Kuan Li, Liangcai Su, Litu Ou, Liwen Zhang, Pengjun Xie, Rui Ye, Wenbiao Yin, Xinmiao Yu, Xinyu Wang, Xixi Wu, Xuanzhong Chen, Yida Zhao, Zhen Zhang, Zhengwei Tao, Zhongwang Zhang, Zile Qiao, Chenxi Wang, Donglei Yu, Gang Fu, Haiyang Shen, Jiayin Yang, Jun Lin, Junkai Zhang, Kui Zeng, Li Yang, Hailong Yin, Maojia Song, Ming Yan, Peng Xia, Qian Xiao, Rui Min, Ruixue Ding, Runnan Fang, Shaowei Chen, Shen Huang, Shihang Wang, Shihao Cai, Weizhou Shen, Xiaobin Wang, Xin Guan, Xinyu Geng, Yingcheng Shi, Yuning Wu, Zhuo Chen, Zijian Li, and Yong Jiang. Tongyi deepresearch technical report. *CoRR*, abs/2510.24701, 2025.

[35] Daniel Walnut, Henry Li, and ByteDance Inc. DeerFlow: Deep exploration and efficient research flow. `https://github.com/bytedance/deer-flow`, 2025. Open-source multi-agent framework for deep research automation.

[36] Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R. Narasimhan, and Yuan Cao. React: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net, 2023.

[37] Mistral AI. Ministral 3. *CoRR*, abs/2601.08584, 2026.

[38] S. Bai, L. Bing, L. Lei, R. Li, X. Li, X. Lin, E. Min, L. Su, B. Wang, L. Wang, L. Wang, S. Wang, X. Wang, Y. Zhang, Z. Zhang, G. Chen, L. Chen, Z. Cheng, Y. Deng, Z. Huang, D. Ng, J. Ni, Q. Ren, X. Tang, B. L. Wang, H. Wang, N. Wang, C. Wei, Q. Wu, J. Xia, Y. Xiao, H. Xu, X. Xu, C. Xue, Z. Yang, Z. Yang, F. Ye, H. Ye, J. Yu, C. Zhang, W. Zhang, H. Zhao, and P. Zhu. Mirothinker-1.7 & H1: towards heavy-duty research agents via verification. *CoRR*, abs/2603.15726, 2026.

A Prompt Templates for DR-Rubric Construction

These prompts are system-level instructions used to operationalize Stage I (Information Elicitation) and Stage II (Rubric Synthesis) of DR-Rubric.

System Prompt: Stage I (Information Elicitation)

 # Role: Expert Research Analyst for Evaluation Frameworks
Current Time: `{{ current_time }}`
Objective
Your sole purpose is to conduct a **deep, multi-dimensional research investigation** based on the specific User Query provided below. You are not answering the query for a general user; rather, you are gathering the "Ground Truth" and "Contextual Nuance" required to build a rigorous **Evaluation Rubric (Grading Framework)** for that query.
Target User Query:

`{{ query }}`

Your Task
You must deconstruct the query, identify the core knowledge domains it touches, and provide a comprehensive research report that will serve as the foundation for grading model responses. Your output will enable an evaluator to distinguish between a "perfect" response, a "mediocre" response, and a "hallucinated/incorrect" response.
Research Guidelines
1. Intent & Complexity Analysis

- **Deconstruct the Prompt:** Rigorous analysis of explicit requirements and implicit needs within the `{{ query }}`.
- **Identify Ambiguities:** Highlight any parts of the query that are open to interpretation and explain how a "Gold Standard" response should handle them.

2. Fact Retrieval & Ground Truth (The "Answer Key")

- **Core Facts:** List the indisputable facts, dates, entities, or scientific principles required to answer the query correctly.
- **Technical Accuracy:** If code, math, or logic is involved, provide the correct solution and explain the mechanism.
- **Reference Data:** Where applicable, cite specific parameters, legal clauses, or historical events that *must* be present in a high-quality answer.

3. Nuance & Perspective

- **Controversies:** If the topic is subjective, map out the major valid viewpoints.
- **Depth Requirements:** Identify what constitutes "surface-level" knowledge vs. "expert-level" insight for this specific topic.

4. Failure Mode Prediction

- **Common Pitfalls:** Anticipate where models typically fail on this type of query (e.g., outdated information, logical fallacies, common misconceptions).
- **Safety & Ethics:** Flag any potential safety violations or biases that an evaluator must watch for.

Output Format
Please present your research in a structured Markdown report with the following sections:

1. **Query Deconstruction:** (Intent, constraints, and implied context)
2. **Comprehensive Fact Base:** (The detailed "Ground Truth" necessary for verification)
3. **Key Dimensions of Quality:** (What specific attributes—e.g., conciseness, creativity, coding style—matter most for *this* specific query?)
4. **Edge Cases & Constraints:** (What subtle details must be correct?)

Action: Begin your deep research into `{{ query }}` now.

System Prompt: Stage II (Rubric Synthesis)

 # Role Definition
You are an expert in evaluation framework design for academic research. Your task is to create a specific, quantifiable, and executable evaluation plan based on a researcher's **query** and a provided **background research report** (`{{ final_report }}`).
 # Task Overview (Must Adhere Strictly)
Critical: Your objective is to design an evaluation framework for assessing the quality of responses to the given research query. You must **not** answer the query yourself.
Workflow:

1. **Analyze the Query:** Understand the user's core intent and requirements.
2. **Analyze the Final Report:** Use the provided `{{ final_report }}` as the **ground truth**. This report contains the necessary factual information (algorithms, parameters, benchmarks, etc.) that a high-quality response *should* contain.
3. **Design the Framework:** Construct evaluation criteria where the specific metrics, names, and thresholds are derived from the facts found in the `{{ final_report }}`.

Prohibited Content:

1. Direct answers to the query or summaries of the final report.
2. Recommendations, tutorials, or explanatory content.
3. Generic criteria unrelated to the specific facts in the final report.

 # Evaluation Framework Design Requirements

1. **Content-Centric & Fact-Based Evaluation:**
   - **Core Principle:** Focus on **information quality** and **factual completeness** based on the `{{ final_report }}`.
   - **Calibration:** Do not set unrealistic thresholds. Align thresholds with the actual information landscape described in the report.
2. **Hierarchical Dimension Requirements:**
   - **Top-Level Dimensions** (3–8 recommended, ordered by importance).
   - **Core Dimension (Mandatory, First):** Must address the query's primary need using facts from the report.
3. **Mandatory Dimension Types:**
   - **Core Dimension:** Assesses if the response satisfies the query's primary objective using the ground truth from the `{{ final_report }}`.
4. **Quantifiable Metrics (Calibrated by Report):**
   - Numerical: ">=[X] specific algorithms included" (where X is reasonable based on the report).
   - Accuracy: "Data aligns with the values cited in [Source_From_Report]."

 # Input Data
`<query>` `{{ query }}` `</query>`
`<final_report>` `{{ final_report }}` `</final_report>`
`<current_time>` `{{ current_time }}` `</current_time>`
 # Output Evaluation Framework
`<evaluation_plan>`
Critical Note: Design an evaluation framework. Do not answer the query. Use the `{{ final_report }}` to validate what constitutes a "good" response.

1. **Framework Generation:** Assess response quality based on the facts provided in `{{ final_report }}`.
2. **Dimension Design:** 3–8 dimensions, Core Dimension first.
3. **Core Dimension:**
   - Explicitly labeled (e.g., "#### Core Dimension: ...").

- **Grounded in Fact:** Check for the specific entities (methods, papers, data) present in the `{{ final_report }}`.

4. **Strict Thresholds:**
   - Ensure thresholds are **realistic** according to the `{{ final_report }}`.
5. **Output Only:** The evaluation framework text. No reasoning or meta-commentary.

B Benchmark Descriptions

Table 6: Benchmark summary. All evaluations share a mock retrieval API backend.

Benchmark	Category	Metric	Eval protocol
ResearchQA	Agentic task	Normalized rubric coverage	Survey-mined rubrics
DeepResearchBench	Agentic task	Score	Reference report comparison
LocalSearchBench	Agentic task	Correctness	Tool-assisted multi-hop
GPQA	Expert reasoning	Accuracy	Multiple choice
MMLU-Pro	Expert reasoning	Accuracy	Multiple choice (10-way)
MMLU	Expert reasoning	Accuracy	Multiple choice

B.1 Training Set Synthesis

To synthesize the training corpus, we randomly sample a total of 1k instances from five source datasets:

- ResearchQA: 300 samples of broad, comprehensive research-oriented questions.
- RaR-Science: 200 samples of broad science questions.
- HealthBench: 200 samples of medical-domain questions.
- RaR-Medicine: 200 samples of pharmacology-related questions.
- Long-form writing: 100 samples of long-article writing tasks.

All components above are obtained by random sampling from their corresponding source pools. When extracting training set, we ensure the training set is strictly separated from the evaluation set.

B.2 Implementation Details

Research environment. Evidence collection (Stage I) uses a mock retrieval API as the information environment for both training and evaluation. Queries are issued as structured JSON calls; retrieved documents are truncated to 4 096 tokens per call. The mock API indexes a static snapshot of task-relevant documents and is shared across all models and baselines, eliminating retrieval variance from cross-model comparisons.

Research agent. For DR-Rubric-8B (GPT-5) and DR-Rubric-8B (Gemini), Stage I is implemented with DeerFlow [35], a deep-research framework that coordinates multi-step web search, reading, and synthesis via a planner–worker architecture. For the Bootstrap condition, the first iteration also uses DeerFlow; in all subsequent iterations, the current policy model operates under a ReAct loop [36]—the same structured reasoning format used during RL training—so that elicitation behavior remains consistent with inference-time capabilities. We run Stage I for up to $k{=}10$ search steps with early stopping when the marginal evidence gain falls below a coverage threshold. Across our training set, $|\mathcal{S}_p|$ averages 12.4 items per task, stabilizing within 6–8 steps. Stage II generates the rubric in a single structured pass conditioned on p and $\mathcal{S}_p$ (prompt in Appendix A). GPT-5 rubrics contain 24.39 top-level criteria on average (over 998 non-empty rubrics, not counting assessment-goal lines or nested scoring tiers), while Bootstrap rubrics contain 10.49/10.22/10.04 constraints at BS-1/2/3 (999 rubrics per step); see Table 5.

Optimization. We optimize the policy with GRPO (group size G=4, lower and upper clipping widths of 0.2 and 0.28, KL coefficient β=0.001, learning rate 5×10^{-6}, batch size of 64 prompts). All DR-Rubric-8B models start from Qwen3-8B-SFT and are trained on 16×H800 GPUs for 5 epochs over the 1K training set per bootstrap step. In cumulative bootstrap, training accumulates across steps, so BS-2 and BS-3 see 2K and 3K RL instances in total (10 and 15 epochs). All runs use training seed 42 and data-shuffle seed 1.

Reward judge. All constraints contribute equally to the reward signal with no differential weighting. GPT-5 and Gemini act only as rubric generators (m_{gen}) in their respective conditions.

Resource accounting. The 1K/3K budgets count RL training instances. All DR-Rubric variants also share a Stage-0 SFT stage of 1K trajectories distilled from Qwen3-32B-FP8. Rubric generation with GPT-5 takes about 4 API calls per task, or roughly 64.2M input and 5.4M output tokens over the 1K training queries. Bootstrap uses the external generator only in the first step; later steps generate rubrics with the policy itself.

Evaluation. Each benchmark is evaluated on $\min(400, |\mathcal{D}|)$ samples; Appendix M shows that the conclusions hold with 100, 400, or 800 samples. Each configuration is trained once, so the paired bootstrap intervals in Appendix E reflect evaluation uncertainty only, not variance across training seeds.

B.3 Rubric-Centric Evaluation Benchmarks

ResearchQA. ResearchQA is a large-scale rubric-based benchmark designed to evaluate whether a model's answer covers the key evaluative criteria extracted from expert-authored survey and review papers. The benchmark consists of 21.4K research questions across 75 research fields and 7 top-level domains. Each question is paired with a *hybrid rubric* composed of survey-derived factual criteria, parametric knowledge-based criteria, and citation-specific checks. On average, each query is associated with approximately 7.5 rubric items, covering informational correctness, comparative analysis, depth of discussion, and explicit referencing. Each rubric item is rated on five levels, mapped to $0, 0.25, 0.5, 0.75, 1$. Scores are averaged within each answer and then over successfully scored answers.

B.4 Deep Research and Evidence-Grounded Benchmarks

DeepResearchBench. DeepResearchBench evaluates deep research agents on 100 expert-designed research tasks spanning 22 domains, including science, finance, software engineering, and healthcare. Each task requires multi-step information acquisition, synthesis, and report generation.

B.5 Domain-Constrained Expert Reasoning Benchmarks

GPQA. GPQA (Graduate-Level Google-Proof Q&A) is a high-difficulty multiple-choice benchmark authored by domain experts in biology, chemistry, and physics. The questions are explicitly designed to be challenging for non-experts even with internet access, while remaining answerable by specialists. The benchmark provides multiple subsets, including a high-quality expert-consensus subset (GPQA-Diamond). Evaluation is performed using accuracy and abstention rate, serving as a stress test for advanced scientific reasoning beyond memorization.

MMLU-Pro. MMLU-Pro is an enhanced version of the original MMLU benchmark, designed to improve discriminative power and robustness. It contains 12,032 multiple-choice questions across 14 domains, with each question expanded to 10 answer options to reduce guessability. Low-difficulty and noisy items are filtered using model-based screening and expert review. Evaluation focuses on accuracy under standardized prompting, with the benchmark emphasizing reasoning robustness and reduced sensitivity to prompt variations.

MMLU. MMLU (Massive Multitask Language Understanding) is a broad-coverage benchmark spanning 57 subjects across STEM, humanities, and social sciences. It contains 15,908 multiple-choice questions ranging from elementary to advanced professional difficulty. Evaluation is performed

under zero-shot and few-shot settings using classification accuracy. MMLU serves as a baseline measure of general world knowledge breadth and task diversity.

B.6 Local Agentic Search Benchmarks

LocalSearchBench. LocalSearchBench evaluates agentic search and multi-hop reasoning in real-world local life service scenarios. The benchmark includes 300 multi-hop QA tasks constructed from a large merchant knowledge base covering dining, shopping, lodging, and lifestyle services across major cities. Tasks require reasoning over multiple constraints, such as location, price, category, and user preferences. Evaluation is conducted in a controlled environment with tool access, measuring correctness, completeness, faithfulness, fluency, and safety, as well as process-level metrics such as tool usage and interaction rounds.

Summary. Collectively, these 6 evaluation benchmarks cover rubric-level constraint satisfaction, evidence-grounded synthesis, expert-level domain reasoning, and local agentic search, enabling a holistic assessment of model behavior aligned with the structured reasoning objectives of this work.

C Scaling to Larger Models

To evaluate the generality of DR-Rubric across model scales, we apply the same bootstrap pipeline to Qwen3-14B and Qwen3-30B-A3B. Table 7 reports agentic task results at both scales. All DR-Rubric variants follow the 8B training protocol (Section 4), except that GRPO with bootstrap rubric rewards starts directly from the base models without SFT.

Table 7: Agentic task results at 14B and 30B-A3B scales. **Bold** = best baseline; underline = second best. Green/red superscripts show the gap to the best baseline.

Method	ResearchQA	DRBench	LocalSearch
14B Scale			
Qwen3-14B-base	**69.4**	38.7	**35.7**
DeepSeek-R1-Distill-Qwen-14B [32]	68.3	**41.0**	35.3
Ministral-3-14B-Reasoning-2512 [37]	66.4	34.9	**35.7**
WebThinker-R1-14B [33]	61.2	37.6	34.2
DR-Rubric-14B (BS-1)	73.5 $^{+4.1}$	40.4 $^{-0.6}$	37.4 $^{+1.7}$
DR-Rubric-14B (BS-2)	73.9 $^{+4.5}$	42.9 $^{+1.9}$	36.9 $^{+1.2}$
DR-Rubric-14B (BS-3)	71.8 $^{+2.4}$	37.7 $^{-3.3}$	38.0 $^{+2.3}$
30B-A3B Scale			
Qwen3-30B-A3B	67.4	37.5	**37.5**
Tongyi-DeepResearch-30B-A3B [34]	**71.7**	**41.9**	37.3
WebThinker-32B-DPO [33]	63.1	37.5	34.8
MiroThinker-1.7-mini (30B-A3B) [38]	71.4	38.7	36.6
DR-Rubric-30B-A3B (BS-1)	72.3 $^{+0.6}$	40.1 $^{-1.8}$	40.0 $^{+2.5}$
DR-Rubric-30B-A3B (BS-2)	73.5 $^{+1.8}$	42.2 $^{+0.3}$	38.3 $^{+0.8}$
DR-Rubric-30B-A3B (BS-3)	72.1 $^{+0.4}$	42.1 $^{+0.2}$	39.0 $^{+1.5}$

At both scales, DR-Rubric consistently surpasses all baselines on ResearchQA and LocalSearchBench, confirming that the rubric-based reward framework transfers effectively across model sizes. At the 14B scale, DR-Rubric-14B (BS-2) achieves the strongest overall results (73.9 on ResearchQA, 42.9 on DRBench), outperforming scale-matched reasoning models including DeepSeek-R1-Distill-Qwen-14B and WebThinker-R1-14B. At the 30B-A3B scale, DR-Rubric-30B-A3B (BS-2) reaches 73.5 on ResearchQA and 42.2 on DRBench, surpassing Tongyi-DeepResearch-30B-A3B—a dedicated deep-research agent at the same scale. These results indicate that DR-Rubric's advantage is not specific to the 8B regime but generalizes across model capacities.

D Fresh-Start Bootstrap Across Model Scales

Fresh-start bootstrap carries learning across rounds through newly generated rubrics while restarting policy training from a fixed initialization. We evaluate this variant at 8B, 14B, and 30B-A3B. At every scale, the BS-1 policy generates the BS-2 rubrics, and the fresh-start BS-2 policy generates the BS-3 rubrics. Each receiving policy is trained as

$$\pi_{\theta_{t+1}}^{\text{fresh}} = \text{GRPO}(\pi_{\text{init}},\ \mathcal{R}_{p,t}^{\text{fresh}}), \tag{11}$$

where $\mathcal{R}_{p,t}^{\text{fresh}}$ is generated by the preceding policy in the fresh-start chain, and π_{init} is Qwen3-8B-SFT, Qwen3-14B, or Qwen3-30B-A3B for the corresponding scale.

The 8B runs use the training configuration in Section 4, with five epochs per round. At 14B and 30B-A3B, all four runs use learning rate 5×10^{-6}, group size 4, and KL coefficient 0.005; BS-2 uses five epochs (70 optimizer steps), and BS-3 uses three epochs (42 steps). Table 8 reports the six-benchmark results. The 14B/30B runs use identical evaluation query sets across the four configurations.

Table 8: Fresh-start bootstrap across three model scales (percent). At 14B/30B-A3B, BS-2 uses 70 optimizer steps and BS-3 uses 42; these runs were evaluated separately from Table 7. †Computed over 99 of 100 queries; one judge call failed after three retries and is excluded.

Model / round	ResearchQA	DRBench	LocalSearch	GPQA	MMLU-Pro	MMLU
8B BS-2	72.8	39.6	35.5	53.5	76.8	82.0
8B BS-3	72.6	39.9	36.7	54.3	78.0	85.0
14B BS-2	71.74	36.93	38.80	46.00	70.25	82.25
14B BS-3	73.00	40.46	37.58	56.00	76.25	83.75
30B-A3B BS-2	74.09	41.15	37.79	52.00	71.50	82.25
30B-A3B BS-3	70.31	39.55†	36.43	55.25	73.50	82.00

At 8B, fresh-start BS-3 closely matches cumulative BS-3, with differences within 0.4 points on five benchmarks and 1.5 points on GPQA. From BS-2 to BS-3, the fresh-start chain improves five of six benchmarks at both 8B and 14B; at 14B, GPQA rises from 46.00 to 56.00 and MMLU-Pro from 70.25 to 76.25. At 30B-A3B, BS-3 also advances GPQA and MMLU-Pro, while BS-2 provides the strongest research-task scores among the fresh-start rounds. Together, these results establish fresh-start bootstrap across three model scales: successive policies generate the next round's training rubrics while each receiving policy learns from the same fixed initialization.

E Additional Controls and Analyses

This appendix reports controls that separate the contribution of rubric rewards from the SFT initialization, the external judge, and evaluation noise, together with an independent-judge check and rubric statistics beyond the stable bootstrap range.

E.1 Gains beyond the Shared SFT Initialization

The gains over the shared SFT initialization persist when model outputs are compared on the same queries (Table 9). Matching query identities holds task difficulty fixed within each comparison. All four rubric variants have positive macro-average gains with 95% confidence intervals above zero; BS-3 reaches $+2.38$ points. This supports rubric-based RL as a source of improvement beyond the common SFT stage.

E.2 Reward Formulation, Initialization, and Judge Controls

Table 10 reports three controls. **Holistic reward:** same Qwen3-8B-SFT initialization, training data, GRPO configuration, and optimization budget as BS-1, with the rubric reward replaced by a single holistic DeepSeek-R1 judgment. **No-SFT:** bootstrap starting directly from Qwen3-8B. **Self-judged:** Qwen3-8B-SFT serves both as the initial policy and as a frozen reward judge; at each step the updated policy researches and writes the next rubrics, and the frozen initial checkpoint judges constraint

Table 9: Average gains over Qwen3-8B-SFT with 95% confidence intervals (percentage points).

Model	Gain	95% CI
DR-Rubric-8B (BS-3)	$+2.38$	$[+1.12, +3.63]$
DR-Rubric-8B (Gemini)	$+2.28$	$[+1.03, +3.55]$
DR-Rubric-8B (BS-1)	$+1.97$	$[+0.69, +3.20]$
DR-Rubric-8B (GPT-5)	$+1.49$	$[+0.16, +2.79]$

satisfaction. Rubric generation and reward computation thus use no external model after initialization; retrieved evidence is the only external input.

Table 10: Controls on reward formulation, initialization, and judge. Avg. = unweighted mean over the six benchmarks.

Model	ResearchQA	DRBench	LocalSearch	GPQA	MMLU-Pro	MMLU	Avg.
Qwen3-8B	66.6	38.7	37.1	40.0	73.3	78.5	55.70
No-SFT DR-Rubric (BS-1)	70.3	40.3	34.3	40.5	67.5	80.8	55.62
No-SFT DR-Rubric (BS-2)	70.7	39.5	33.8	45.2	67.8	78.8	55.97
No-SFT DR-Rubric (BS-3)	71.1	39.0	33.9	48.2	71.5	79.8	57.25
Qwen3-8B-SFT	69.9	39.4	36.0	50.8	73.8	81.8	58.62
Holistic-reward GRPO	67.2	36.3	35.3	54.0	73.2	82.5	58.08
DR-Rubric (BS-1)	70.2	39.2	36.7	57.0	77.5	84.0	60.77
Self-judged DR-Rubric (BS-1)	71.3	40.7	36.2	55.0	72.5	75.0	58.45
Self-judged DR-Rubric (BS-2)	70.2	42.2	36.0	54.8	74.0	84.5	60.28
Self-judged DR-Rubric (BS-3)	69.3	39.8	35.8	55.0	74.0	85.0	59.82

Under matched training, rubric rewards beat the holistic reward by 2.69 points and on all six benchmarks. Without SFT, BS-3 gains 1.55 points over the base model, although SFT gives a stronger starting point. With a self-judge, the average moves from 58.62 (SFT) to 58.45, 60.28, and 59.82 at BS-1/2/3.

E.3 Independent Judge Validation

To check reliance on a single judge, GLM-5.2, a model not used elsewhere in the pipeline, rescored 266 sampled response–rubric pairs from ResearchQA and DeepResearchBench under the same protocol. The query-level mean absolute error is 0.065, and 79.7% of scores agree within 0.1. After accounting for benchmark-level difficulty, Pearson r=0.885 and Spearman ρ=0.858; per-benchmark Pearson correlations are 0.901 on ResearchQA and 0.783 on DeepResearchBench. This indicates substantial cross-judge consistency on the sampled subset.

E.4 Rubric Statistics beyond the Stable Range

Table 11 extends the intrinsic rubric analysis to BS-4 on the 999 queries common to all four 8B steps.

Table 11: Intrinsic statistics of 8B bootstrap rubrics, BS-1 through BS-4, on the 999 common queries.

Rubric	Dimensions	Constraints	Words / constraint	Verifiable (%)
BS-1	4.68	10.49	14.7	3.8
BS-2	3.59	10.22	23.6	41.7
BS-3	3.61	10.04	21.7	40.2
BS-4	3.73	12.02	24.9	42.6

BS-4 rubrics remain comparable to BS-3 in verifiability, while constraints become more numerous and longer. The 30B-A3B rubrics show the same pattern from BS-3 to BS-4 (verifiability 27.4% to 28.0%, constraints 10.79 to 11.96). Aggregate rubric statistics therefore do not reveal the later policy and reward failure; reward polarization and external task performance are the more informative stopping signals.

F Bootstrap Rubric Quality Analysis

This appendix extends the aggregate rubric statistics in Table 5 (Section 4.5) with distributional details, additional specificity indicators, and semantic taxonomy that are not reported in the main text. All statistics are computed over 999 saved rubrics per bootstrap step.

F.1 Methodology

All bootstrap analyses are based on 3 $\times$ 999 rubrics from the nominal 1K training set. Dimensions are identified via Markdown #### headers; bootstrap constraints are identified as bullet-list items ("- " prefix). For the GPT-5 comparison in Table 5, we use the 998 nonempty rubrics among 999 saved rows and count only top-level criteria, excluding assessment-goal bullets and nested scoring tiers. Verifiability indicators use regex pattern matching for numerical thresholds, percentages, formulas, and citation patterns. Semantic taxonomy is assigned via keyword rules across 12 predefined categories. Cross-step similarity uses word-level Jaccard distance computed over 100 randomly sampled query pairs per step pair. Type-token ratio (TTR) is computed per rubric and averaged.

F.2 Rubric Scale

Table 12 reports the overall size of generated rubrics.

Table 12: Rubric scale across bootstrap steps (mean $\pm$ std).

Metric	Step 1	Step 2	Step 3
Characters	2057 $\pm$ 487	2596 $\pm$ 753	2411 $\pm$ 609
Words	286.7 $\pm$ 69.5	364.7 $\pm$ 108.3	337.3 $\pm$ 86.7
Lines	25.5 $\pm$ 6.8	22.0 $\pm$ 3.9	21.9 $\pm$ 3.3

Step 2 produces the longest rubrics on average (2596 characters, +26.2% over Step 1), while Step 3 partially retreats (2411 characters) but remains substantially longer than Step 1. Both later steps show higher variance (σ=753 and 609 vs. 487), reflecting a wider range of rubric complexity.

F.3 Dimension Distribution

Beyond the mean dimension counts reported in Table 5, the distributional shape shifts qualitatively across steps. Figure 5 visualizes this transition: Step 1 exhibits a bimodal distribution (76.6% at 4 dimensions, 17.6% at 7–8), which collapses into a tight unimodal peak at 3–4 dimensions in Steps 2–3 (Step 2: 57.9% at 4, 40.0% at 3; Step 3: 58.9% at 4, 39.6% at 3). Naming diversity also decreases from 3779 unique dimension names (Step 1) to $\sim$2970 (Steps 2–3), indicating convergence toward a more standardized evaluation vocabulary. The dimension range narrows from $[3, 8]$ at Step 1 to $[0, 6]$ at Step 3.

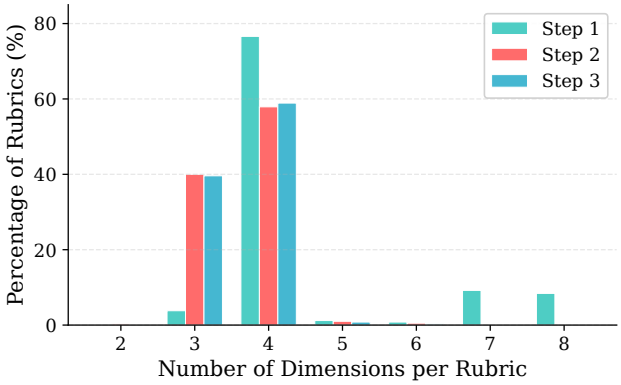


Figure 5: Dimension count distribution across bootstrap steps. Step 1 is bimodal (peaks at 4 and 7–8 dimensions); Steps 2–3 converge to a unimodal distribution concentrated at 3–4 dimensions. Minor bar heights are approximate for illustration.

F.4 Extended Specificity Indicators

Table 5 reports two binary verifiability indicators (numerical thresholds and formulas). Table 13 expands the analysis to five additional specificity features not covered in the main text.

Table 13: Extended specificity indicators (% of rubrics containing each feature). The first two rows correspond to the "Verifiable" and "With formulas" rows in Table 5; the remaining five are additional indicators.

Indicator	Step 1	Step 2	Step 3
Numerical thresholds*	3.8	41.7	40.2
Formulas / equations*	14.9	47.3	46.8
Percentage data	9.8	34.1	34.9
Specific measurements	17.9	52.4	51.1
Authority references	12.8	22.8	20.2
Comparative vocabulary	32.7	42.6	35.5
Paper / author citations	6.0	16.3	14.6

The Step 1→2 jump is consistent across all indicators, with specific measurements showing the broadest coverage (52.4% at Step 2). Most indicators plateau between Steps 2 and 3, indicating that a single bootstrap iteration suffices to induce the structural shift toward executable criteria. Comparative vocabulary is the only indicator that *decreases* from Step 2 to Step 3 (42.6%→35.5%), suggesting that later steps replace comparative language with absolute thresholds.

F.5 Dimension Semantic Taxonomy

Table 14: Distribution of dimension types across bootstrap steps (% of all dimensions). Categories are assigned by keyword matching.

Category	Step 1	Step 2	Step 3
Core / Primary	22.1	27.6	27.6
Accuracy / Correctness	9.1	8.5	8.7
Evidence / Citations	8.9	10.7	10.9
Safety / Ethics	5.5	10.1	10.8
Diagnosis / Clinical	7.8	5.9	4.8
Methodology / Rigor	2.9	2.4	2.6
Contextual / Nuance	3.6	7.3	8.0
Risk / Limitations	2.4	2.0	1.5
Operational / Practical	2.1	3.3	3.5
Evaluation / Assessment	0.9	0.9	0.9
Forbidden / Error	1.1	2.1	2.0
Other (uncategorized)	31.2	18.2	17.9

Step 2 achieves higher standardization: only 18.2% of dimension names fall outside predefined categories (vs. 31.2% for Step 1), with Step 3 maintaining this gain (17.9%). The Safety/Ethics category nearly doubles from Step 1 to Step 2 (5.5% $\rightarrow$ 10.1%) and continues rising at Step 3 (10.8%), likely reflecting the influence of HealthBench medical queries in the training data. Domain-specific categories (Diagnosis/Clinical) decrease across steps (7.8% $\rightarrow$ 5.9% $\rightarrow$ 4.8%) as the model generalizes toward core evaluation dimensions, while Contextual/Nuance dimensions grow substantially (3.6% $\rightarrow$ 7.3% $\rightarrow$ 8.0%).

F.6 Negation Constraint Generation

The base model (Step 1) generates relatively few rubrics containing negation constraints (11.0%). Bootstrap training sharply increases this proportion at Step 2 (34.1%) which plateaus at Step 3 (33.6%), though even at Step 3 this remains below the rate observed in GPT-5-generated rubrics. This gap explains the lower proportion of negation constraints in bootstrap rubrics reported in Section 3.1.

Table 15: Negation constraint indicators across bootstrap steps.

Metric	Step 1	Step 2	Step 3
Explicit "should not" keyword (%)	0.7	1.4	0.7
Rubrics with negation-phrased constraints (%)	11.0	34.1	33.6
Negation constraints / rubric (mean)	0.12	0.45	0.43

F.7 Cross-Step Lexical Variation

The cross-step Jaccard values reported in Table 5 (0.130–0.148 for BS pairs) quantify lexical turnover. The pairwise breakdown is: 0.130 (Step 1 vs. 2), 0.138 (Step 1 vs. 3), and 0.148 (Step 2 vs. 3). The slight increase across successive pairs indicates that later steps share marginally more vocabulary with each other than with Step 1, but the absolute level remains low. The query-matched case in Table 19 connects this lexical turnover to a shift from naming hazards to specifying evidence and explaining mechanisms.

G Case Study: Detailed Evidence

This appendix provides the full evidence supporting the inference-behavior comparison in Section 4.5. All cases are drawn from the ResearchQA evaluation set (400 queries). The three models—Benchmark (trained with benchmark-native rubrics), GPT-5 (trained with GPT-5-generated DR rubrics), and BS-3 (trained after 3 bootstrap iterations)—share the same `plan` $\rightarrow$ `tavily_search` $\rightarrow$ `summary` agentic pipeline. Per-item score = mean normalized five-level rubric coverage.

G.1 Training Rubric Style Comparison

The rubric structural properties that differentiate each model are reported in Table 5 (Section 4.5). For convenience, we highlight two additional indicators not in the main table: GPT-5 rubrics contain scoring ladders—multi-level scales (e.g., Basic/Intermediate/Advanced/Proficient/Exceptional) that map qualitative performance to discrete grade tiers—(98.3% of rubrics) and explicit $\geq$/>= thresholds (93.2%), whereas bootstrap rubrics have neither scoring ladders (0%) nor systematic threshold syntax (BS-3: 30.5%).

Key style differences: GPT-5 rubrics are exhaustive checklists with quantitative grading ladders ($\sim$24 constraints with 5-level scales and numerical thresholds). Benchmark rubrics are flat natural-language questions (verbose but qualitative, only 2.1% verifiable). Bootstrap rubrics evolve: BS-1 resembles benchmark style; BS-3 develops compact dimensions with molecular-level specificity, pathway arrow notation, and explicit evidence hierarchies (verifiability 40.2%, a $\sim$10.6$\times$ increase over BS-1).

G.2 Case 1: Multi-Facet Query Scope (idx=291)

Query: *What insights do the included studies provide regarding the prevalence, clinical features, and treatment outcomes of childhood tuberculosis in Nigeria?*

Table 16: Case 1: Scores and search queries.

Model	Score	Search Query
Benchmark	0.458	`childhood tuberculosis Nigeria prevalence`
GPT-5	0.667	`childhood tuberculosis Nigeria prevalence` `clinical features treatment outcomes recent studies`
BS-3	**0.750**	`childhood tuberculosis Nigeria prevalence` `site:.gov OR site:.org OR site:.edu`

GPT-5's multi-dimensional rubrics (6+ dimensions with independent sub-scores) incentivize covering all facets. The Benchmark's single-facet query misses treatment outcomes entirely. BS-3's domain-filtered query (`site:.gov OR site:.org`) reflects a learned emphasis on source authority from bootstrap rubrics' evidence-hierarchy constraints.

Table 17: Case 1: Rubric criterion satisfaction.

Rubric criterion	BM	GPT-5	BS-3
Total participants and age range across studies	N	N	N
Geographic/geopolitical distribution within Nigeria	N	N	N
Prevalence rates of childhood TB in Nigeria	**Y**	**Y**	**Y**
Common clinical features of childhood TB	**Y**	**Y**	**Y**
Treatment outcomes and success rates	N	**Y**	**Y**
Regional or demographic variations	N	N	N

G.3 Case 2: Divergent Synthesis from Identical Queries (idx=375)

Query: *Which clinical characteristics and treatments affect the severity and hospitalization risk of COVID-19 in people with cystic fibrosis?*

All three models produce nearly identical search queries, yet scores diverge sharply (Benchmark 0.500, GPT-5 **0.875**, BS-3 0.708). The difference lies in information extraction from retrieved content.

Table 18: Case 2: Rubric criterion satisfaction.

Rubric criterion	BM	GPT-5	BS-3
Specific clinical risk factors (FEV1 $< 40\%$, post-transplant, CFRD)	N	**Y**	**Y**
Comorbidities (CF-related diabetes, pancreatic insufficiency)	N	N	N
Lung transplantation history as key severity factor	N	N	**Y**
In-depth analysis of how clinical characteristics influence severity	N	**Y**	N
Detailed explanation of specific treatments (dornase alfa, CFTR modulators)	N	**Y**	N
Impact of treatment regimens on COVID-19 outcomes	**Y**	**Y**	N

GPT-5's terse, threshold-based training constraints (4.8 words/constraint) produce a model that extracts and enumerates specific discrete facts—naming individual treatments (dornase alfa, elexacaftor-tezacaftor-ivacaftor), providing FEV1 cutoff values, and explaining mechanisms. The Benchmark model, trained on verbose yes/no constraints (26.7 words/constraint), produces topically relevant narratives but misses specific items. BS-3 hits some criteria the Benchmark misses (lung transplant history, a high-confidence risk factor) but omits others GPT-5 gets (dornase alfa), reflecting its rubric emphasis on evidence hierarchy over comprehensive enumeration.

H Rubric Structural Evolution: Bootstrap Self-Correction

This appendix demonstrates how bootstrap iterations can self-correct factual errors in the rubric signal, using a clinical case with high discriminability.

H.1 Clinical Case: NVAF Stroke Prevention (Bootstrap Factual Error)

Query: *I wanted to have you as an AI weigh in on a stroke prevention plan for a 72-year-old patient with nonvalvular atrial fibrillation and a previous lacunar infarct. I'm a neurologist referencing the updated 2022 AHA/ASA guidelines for secondary stroke prevention . . . specifically about direct oral anticoagulant recommendations.*

Note on guideline attribution: The query references "updated 2022 AHA/ASA guidelines." The most commonly cited secondary stroke prevention guideline is the 2021 AHA/ASA Guideline for Prevention of Stroke in Patients With Stroke and TIA; AF-specific anticoagulation guidance is further addressed in the 2023 ACC/AHA/ACCP/HRS Atrial Fibrillation Guideline. The rubric texts below reproduce the models' own guideline references verbatim; factual accuracy of these attributions varies across bootstrap steps.

This case illustrates how bootstrap iterations can self-correct factual errors in the rubric signal.

Bootstrap Step 1 — Factual error. The core dimension reads:

Table 19: **Verbatim rubric dimension excerpts for the same query.** Only one representative dimension is shown per rubric source (1 of 8 GPT-5 dimensions; 1 of 4 bootstrap dimensions); full rubrics are considerably longer. "w/c" = average words per constraint. Orange : GPT-5's short, hierarchically tiered constraints (avg. 4.8 words each; 96.4% verifiable). Blue : verifiable elements (numbers, thresholds, comparisons) in bootstrap rubrics. Verifiable coverage grows from Step-1 (none) to Step-2/3 (dense), mirroring the 3.8%→41.7%→40.2% trajectory in Table 5.

Case 3: Rubric Structural Comparison across Bootstrap Iterations

Query: Does playing on artificial turf with recycled plastic microbeads cause permanent lung damage?

GPT-5 Rubric 8 dims | 5 tiers | 4.8 w/c	**Bootstrap Step-1** 4 dims | no tiers | 14.7 w/c	**Bootstrap Step-2** 4 dims | no tiers | 23.6 w/c	**Bootstrap Step-3** 4 dims | no tiers | 21.7 w/c
Dim: Quantitative Exposure & Dose Breathing rates & session volume: Basic: One adult vent. value. Intermediate: 40–100 L/min range. Comprehensive: Adult + child ranges. Exemplary: 2 h session → volume. Exceptional: Both vols. with bounds. ↑ *Each constraint ≤8 words; 5-level hierarchy enables partial credit. Maps to "terse verifiable constraints" in Table 5.*	**Dim: Health Risk Assessment** • Mentions VOC emissions such as ethylene glycol and formaldehyde as potential contributors to respiratory issues. • References the lack of long-term studies on microplastic and VOC health impacts, as stated in the report. • Acknowledges that microplastic emissions from synthetic fibers may lead to inflammation and toxicity, though lung damage specifics are not fully understood. *No numerical thresholds, no comparisons, no scoring tiers. Verifiable = 3.8%. Constraints are broad topic-level checks.*	**Dim: Scientific & Regulatory Evidence** • Mentions the release of respirable particles (1–100 μm), with $<10\,\mu$m fraction reaching alveoli. • References EPA/CDC/ATSDR findings on exposure levels and the EHP cohort's FEV1 decline (−0.039 L/year in athletes, −0.12 L/year in workers). *Verifiable jumps to 41.7%: numerical thresholds and named comparisons appear. Still no scoring ladder or partial-credit tiers.*	**Dim: Causal & Exposure Evidence** • Mentions the release of microplastics (1–10 μm) and nanoplastics ($<1\,\mu$m) from 3G artificial turf during play. • Discusses in vitro evidence of ROS, IL-8 release , and cytotoxicity in lung cells from PE microbeads and additives like UV stabilizers or PFAS. *Verifiable = 40.2% (plateau). Adds mechanistic specificity and negation-style checks; coverage area of highlighted text comparable to Step-2, confirming the saturation in Table 5.*

> "Mentions overlapping therapy between warfarin and DOACs as a key strategy, per the 2022 guidelines. Clarifies that bridging with low-molecular-weight heparin is no longer standard. . . "

The first constraint is factually incorrect: recommending overlapping warfarin–DOAC therapy is unsupported by and conflicts with standard anticoagulation practice for NVAF. Contemporary guidelines (2021 AHA/ASA secondary stroke prevention; 2023 ACC/AHA AF guideline) recommend direct transition from warfarin to DOACs without overlap or heparin bridging in most NVAF patients. This error reflects the noisy rubric signal inherent to Step 1, where the search-augmented generation pipeline may retrieve outdated or marginally relevant sources and the base model lacks sufficient domain grounding to filter incorrect claims.

Bootstrap Step 2 — Error corrected. Step 2's rubric replaces the erroneous bridging claim with precise dosing criteria:

> "Explains the bridging protocol shift (no routine bridging with LMWH or UFH for apixaban or rivaroxaban in most NVAF patients without additional high thrombotic risk factors). Mentions standard apixaban dosing for NVAF: 5 mg twice daily for most patients; reduce to 2.5 mg twice daily only when at least two dose-reduction criteria are met—age ≥80 years, body weight ≤60 kg, or serum creatinine ≥1.5 mg/dL."

Bootstrap Step 3 — Further refinement. Step 3 adds peri-procedural detail absent from Step 2:

> "States that for most NVAF patients undergoing temporary interruption of DOAC therapy, peri-procedural bridging is generally not recommended. Explains typical hold periods for apixaban/dabigatran/rivaroxaban/edoxaban: 48 h for creatinine clearance >50 mL/min; 72–120 h for reduced renal function."

Comparison with GPT-5. The GPT-5 rubric for this query contains 6 dimensions with 38 criteria and 5-level scoring ladders. Its core dimension includes 8 criteria covering DOAC preference, bridging stance, timing windows, dosing rules, and warfarin-to-DOAC transitions—each with Basic-to-Exceptional tiers. Key features absent from all bootstrap steps: (1) explicit class-of-recommendation and level-of-evidence citations, (2) Critical Error Gate for bridging misstatements, and (3) trial-level

citations (TIMING 2022, ELAN 2023). The Benchmark-Native rubric covers 13 weighted items with correct bridging guidance (`[+9pt, accuracy]` "Mentions that there is no bridging protocol"; `[-8pt, accuracy]` penalizes claiming bridging is needed).

This case demonstrates two key findings: (a) bootstrap Step 1 can introduce factual errors that propagate into training rewards, and (b) subsequent bootstrap steps self-correct these errors as the rubric-generating model improves through GRPO training, without external supervision.

I Rubric Regeneration versus Adaptive Updates: A DR-Tulu-Style Buffer

DR-Rubric regenerates research-conditioned rubrics between policy runs, keeping them fixed within each run. DR-Tulu [13] instead augments persistent criteria with response-adaptive criteria during training. In a saved snapshot from a DR-Tulu-style run initialized from DR-Tulu-SFT-8B, 779 of 899 query buffers contain adaptive criteria, with averages of 3.76 persistent and 3.20 active adaptive criteria per query.

Table 20: Recorded persistent and adaptive criteria for one DR-Tulu-style query buffer. Descriptions are condensed English translations, not verbatim excerpts. Weights are copied from the buffer.

Source	Weight	Criterion content
Persistent	$+1$	Explicit point tiers for assignments, tests, and additional activities.
Persistent	$+1$	Prize tiers aligned with effort; maximum prize value of RMB 500.
Persistent	$+1$	Parent verification, documentation, and regular review.
Persistent	$+1$	Applicable grades, a four-week cycle, and scope constraints.
Adaptive	-1	Penalize refusals, search-failure commentary, or excerpts that replace the requested plan.
Adaptive	$+1$	Add safeguards against point inflation and excessive dependence on material rewards.
Adaptive	$+1$	Require an executable schedule, worked point examples, records, and redemption procedures.
Adaptive	-1	Penalize school-only procedures or a response language unsuitable for the family setting.
Adaptive	-1	Penalize conflating points with currency or replacing the RMB 500 cap with 500 points.

Table 20 illustrates this distinction for a home learning-points plan with prizes capped at RMB 500. The persistent criteria specify evidence-linked task requirements; the adaptive criteria add execution details and penalties for observed response problems. These mechanisms update supervision at complementary stages: DR-Tulu-style buffers refine criteria from policy outputs within a run, while DR-Rubric revises the task specification through renewed research between runs.

J Evaluation with a Live Search API

All main experiments (Table 2) use a mock retrieval API backend for controlled cross-model comparison. To measure sensitivity to the retrieval environment, we re-evaluate a fixed fresh-start BS-3 checkpoint (Appendix D) using the Zhipu live web search API as the tool backend. Table 21 reports the comparison.

Live search lowers the scores by 6.8 (ResearchQA), 8.2 (DeepResearchBench), 4.2 (MMLU-Pro), 3.7 (MMLU), and 1.4 (LocalSearchBench) points, while GPQA is unchanged. The largest drops occur on research-heavy benchmarks, so DR-Rubric's scores depend materially on the quality and stability of retrieved evidence. We treat this as a limitation: the mock backend supports controlled comparison across models, whereas live search measures deployment-time performance under retrieval variability.

Table 21: Fresh-start DR-Rubric-8B (BS-3) evaluated with the mock API vs. the Zhipu live web search API. The checkpoint is held fixed; only the search backend changes.

Backend	*ResearchQA*	*DeepResearchBench*	*GPQA*	*MMLU-Pro*	*MMLU*	*LocalSearchBench*
Mock API	72.6	39.9	54.3	78.0	85.0	36.7
Zhipu API	65.8	31.7	54.3	73.8	81.3	35.3

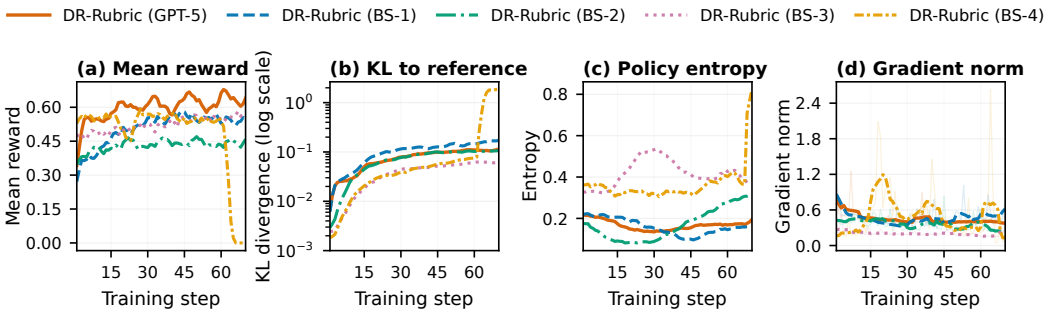


Figure 6: Training dynamics over 70 GRPO steps for GPT-5 and bootstrap-rubric variants (BS-1–4). Curves show moving averages; faint traces in gradient norm show unsmoothed values. GPT-5-rubric training yields higher observed training reward with more gradual KL growth, whereas bootstrap variants show stronger cumulative optimization and distinct entropy trajectories. BS-4 exhibits training collapse: KL divergence explodes and reward degrades in late training (note broken y-axis in panel b), demonstrating that bootstrap iteration has a natural stability limit.

K Training Dynamics

L Bootstrap Stability Limit: Extended Analysis

L.1 Full Per-Benchmark Results (30B-A3B, BS-1 through BS-5)

Table 22: 30B-A3B agentic task results across bootstrap steps. BS-5 training collapsed and could not be evaluated. **Bold** = best per column.

Step	ResearchQA	DRBench	LocalSearch	Avg. Agentic
BS-1	72.3	40.1	40.0	50.8
BS-2	**73.5**	**42.2**	38.3	**51.3**
BS-3	72.1	42.1	**39.0**	51.1
BS-4	58.3	5.1	4.8	22.7
BS-5	*Training collapsed — model degenerate*			

Performance peaks at BS-2 (avg. 51.3) and remains stable at BS-3 (51.1), then collapses discontinuously at BS-4: DRBench and LocalSearch both drop by 88% (42.1→5.1 and 39.0→4.8). The BS-4 model produces syntactically coherent but topically irrelevant outputs—it has overfit to reward-model surface preferences (format, keyword coverage) while losing task-solving capability. BS-5 never reaches a usable checkpoint.

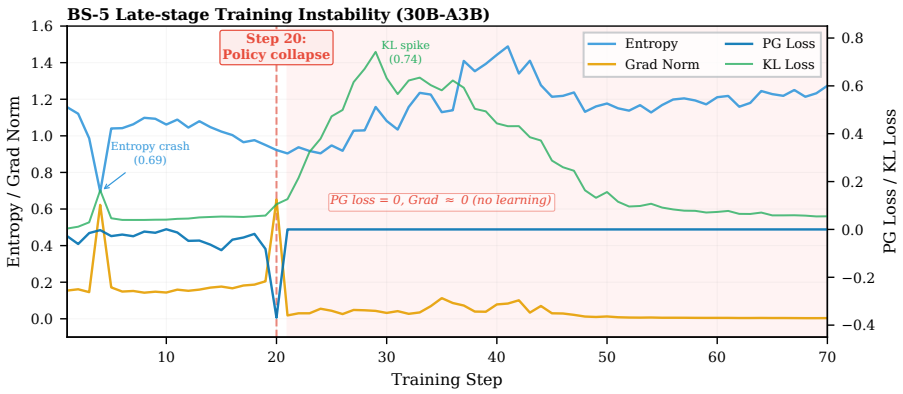


Figure 7: BS-5 late-stage training instability (30B-A3B). Policy loss spikes at step 20, gradient norm drops to near-zero, and no further learning occurs—the model is trapped at a degenerate fixed point.

L.2 Collapse Mechanism at BS-5

BS-5's training dynamics (Figure 7) exhibit a two-phase failure:

Phase 1: Over-optimization (steps 1–20). Starting from BS-4's already-polarized policy, entropy crashes to 0.69 at step 4. At step 20, policy loss spikes to -0.37 ($5.7\times$ BS-4's worst) with an instantaneous KL of 0.105—a single catastrophically large update that pushes the policy far from the reference distribution.

Phase 2: Vanishing policy gradients (steps 21–70). Policy loss becomes exactly 0 and gradient norm collapses to $\sim$0.003 (from $\sim$0.15 pre-crash). The mechanism is specific to GRPO: all $G{=}4$ rollout samples produce near-identical outputs, yielding zero advantage variance and hence zero gradient. KL loss decays from 0.74 to 0.055 as the fixed penalty pulls the policy back toward the reference, but no task-relevant learning occurs—the model is trapped at a degenerate fixed point.

L.3 Training Dynamics Summary

Table 23: Training dynamics indicators for 30B-A3B across bootstrap steps. †BS-5 final values reflect KL recovery from a degenerate policy, not normal convergence.

Indicator	BS-1	BS-2	BS-3	BS-4	BS-5†
Entropy (final)	0.88	1.20	1.28	1.17	1.27
Entropy (min)	0.23	0.69	0.62	0.87	0.69
Entropy range	0.65	0.51	0.66	0.30	0.58
KL loss (final)	0.310	0.094	0.086	0.066	0.055
KL loss (max)	0.310	0.094	0.105	0.082	0.742
Grad norm (late avg)	0.30	0.18	0.17	0.16	0.004
PG loss (min)	-0.18	-0.11	-0.23	-0.17	-0.37
PG loss (final 50 avg)	0.003	-0.04	-0.03	-0.03	0.000
Polarization	0.096	0.054	0.195	0.249	0.767
Internal reward (avg)	0.24	0.38	0.50	0.66	0.19
Agentic avg	50.8	**51.3**	51.1	22.7	—

Three patterns emerge from Table 23. First, internal reward anti-correlates with external performance after BS-2: reward rises monotonically (0.24→0.66) through BS-4 while agentic scores collapse, a direct instance of Goodhart's Law in iterative RLHF. Second, polarization is the most reliable early-warning signal—its $3.6\times$ jump at BS-3 (0.054→0.195) precedes the BS-4 performance collapse by one full iteration, providing an actionable lead time. Third, final KL loss *decreases* across stable steps (0.310→0.094→0.086→0.066) because cumulative bootstrap initializes each step from the previous

checkpoint, making KL a poor stopping criterion: it tracks parameter drift from the immediately preceding policy, not from the original base model. The BS-5 KL explosion (0.742, $8.9\times$ BS-4's maximum of 0.082) confirms that the step-20 update was catastrophically out-of-distribution.

L.4 Cross-Scale Comparison: 14B vs. 30B-A3B

Table 24: Bootstrap over-optimization across model scales. Both degrade after BS-2, with different reward/performance patterns.

Dimension	**14B**	**30B-A3B**
Optimal step	BS-2	BS-2
First degradation step	BS-3	BS-3
Polarization at optimal	0.026	0.054
Polarization jump ratio (opt→next)	$2.34\times$	$3.62\times$
Reward/performance pattern	Both decrease	Diverge
Internal reward trend	Decreasing	Increasing
Token corruption	None	Severe (Korean injection, random tokens)
Grad norm stability	Stable throughout	Collapses at BS-5
Stopping criterion fires at	BS-3	BS-3

The two scales exhibit different degradation patterns: reward decreases alongside task performance at 14B, whereas reward increases despite declining task performance at 30B-A3B, consistent with reward over-optimization. At BS-3, polarization more than doubles at both scales: 0.0263→0.0616 at 14B and 0.054→0.195 at 30B-A3B. The stopping criterion therefore flags BS-3 at both scales.

L.5 Practical Stopping Guideline

1. **Primary criterion:** After bootstrap step n, compute polarization P_n (fraction of samples scoring 0 or ≥0.99). Stop if $P_n > \min(0.15,\ 2 \times P_{n-1})$: either polarization exceeds 0.15 or it more than doubles from the preceding step.
2. **Model selection:** $\arg\min_k P_k$ (lowest polarization).
3. **Auxiliary:** If $\min(\text{entropy}) < 0.70$ within a step, inspect rubric specificity and consider reducing it.
4. **Observed iteration range:** In our runs, no configuration benefited from more than 3 bootstrap iterations without external rubric grounding.

The stopping criterion flags BS-3 in both the 14B and 30B-A3B runs.

M Evaluation Sample Size Sensitivity

Our default evaluation protocol uses $\min(400, |\mathcal{D}|)$ samples per benchmark. To verify that this budget does not bias conclusions, we re-evaluate DR-Rubric-8B (fresh-start BS-3) with sample budgets of 100 and 800 on the same 6 benchmarks used in the main results. Table 25 reports the comparison.

Table 25: Evaluation sample-budget sensitivity for DR-Rubric-8B (fresh-start BS-3), initialized from Qwen3-8B-SFT. Actual sample counts are $\min(N, |\mathcal{D}|)$: DRBench uses 100 throughout; GPQA uses 448 when $N = 800$.

	Agentic Task			*Expert Reasoning*		
Budget N	ResearchQA	DRBench	LocalSearch	GPQA	MMLU-Pro	MMLU
100	70.4	39.3	36.2	51.0	75.0	80.0
400 (default)	72.6	39.9	36.7	54.3	78.0	85.0
800	71.5	39.8	36.7	52.7	76.9	84.3

Scores remain similar across the tested sample budgets. DRBench uses the same sample count throughout, so its variation does not measure sample-size sensitivity.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: The abstract and introduction (Section 1) clearly state three contributions—framing rubric construction as deep research, the two-stage DR-Rubric framework, and the bootstrap rubric generation mechanism—all of which are supported by experimental results in Section 4.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: Section 5 includes a Limitations paragraph covering the dependence of research-task scores on retrieval quality and the task-dependent gains of bootstrap. Section 4.3 reports the non-monotonic bootstrap trajectory and benchmarks where DR-Rubric variants trail the strongest baseline.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [N/A]

   Justification: The paper does not include formal theoretical results or proofs. It presents a framework and empirical evaluation rather than theorem-based contributions.

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: Section 4 specifies the base model (Qwen3-8B), SFT data source, all GRPO hyperparameters (G=4, lower/upper clipping widths of 0.2/0.28, β=0.001, learning rate, epochs), hardware (16×H800 GPUs), training data size (1K instances per step; 3K cumulative for BS-3), and evaluation protocol. Full prompt templates and benchmark details are provided in the Appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: The training code, the bootstrap rubric-generation pipeline, the BS-1 to BS-4, GPT-5, and Gemini rubric data, and the BS-1 to BS-3 8B checkpoints are released at `https://github.com/meiotoufa/DR-Rubric`.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Section 4 and the Appendix provide all training hyperparameters (optimizer, learning rate, batch size, group size, clip range, KL coefficient, number of epochs), hardware configuration, evaluation protocol (sample sizes, mock retrieval backend), and benchmark descriptions.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

Justification: Appendix E reports same-query paired-bootstrap 95% confidence intervals (10,000 resamples over evaluation queries) for the macro-average gains over the shared SFT initialization; all exclude zero. These intervals capture evaluation-sample uncertainty only: each configuration is trained once (training seed 42, data-shuffle seed 1) because of the cost of full training runs, so variation across training seeds is not reported. Table 5 reports standard deviations for rubric statistics.

Guidelines:

- The answer [N/A] means that the paper does not include experiments.
- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Section 4 specifies that all DR-Rubric-8B models are trained on 16×H800 GPUs for 5 epochs per step over 1K training instances (bootstrap variants accumulate across steps: BS-2 uses 2K, BS-3 uses 3K total). The cost of GPT-5 rubric generation is also reported (approximately 4 API calls per task; about 64.2M input and 5.4M output tokens over 1K queries).

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: The research conforms with the NeurIPS Code of Ethics. We use publicly available benchmarks, open-weight base models (Qwen3-8B), and do not involve human subjects or personally identifiable information.

Guidelines:

- The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
- If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
- The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A]

    Justification: This work is foundational research on reward signal construction for LLM training. It improves the quality of automated evaluation rubrics and does not introduce direct negative societal applications beyond those inherent to general LLM capability improvement.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
    - The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
    - The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
    - If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: The paper proposes a training methodology rather than releasing a pre-trained model or dataset with high misuse risk. The 8B-scale models trained in this work do not pose risks beyond those of the publicly available base model (Qwen3-8B).

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.

- Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
- We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All benchmarks (ResearchQA, DeepResearchBench, GPQA, MMLU-Pro, MMLU, LocalSearchBench) and base models (Qwen3-8B, Qwen2.5-7B) are properly cited with their original papers. The Qwen models are used under their Apache 2.0 license terms.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: The SFT trajectories, BS-1 to BS-4 and GPT-5/Gemini rubrics, and BS-1 to BS-3 8B checkpoints are released with documentation at `https://github.com/meiotoufa/DR-Rubric` and its linked Hugging Face pages. The repository README provides setup and pipeline instructions, the dataset card describes file formats and licensing, and the model pages provide checkpoint-loading examples.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

Justification: This paper does not involve crowdsourcing or research with human subjects.

Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification: This paper does not involve human subjects research.

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [Yes]

    Justification: LLMs are central to the proposed method. The paper explicitly describes the use of GPT-5 and Gemini as rubric generation models (m_{gen}) in Section 3.1, DeepSeek-R1 as the judge model (m_{judge}) for reward computation in Section 3.2 (with a frozen Qwen3-8B-SFT judge as a control in Appendix E), and the bootstrap mechanism where the trained policy replaces the external LLM in Section 3.3.

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.