SCOPE2RUBRIC: RECOVERING QUERY-SPECIFIC EVALUATION SCOPE BEFORE RUBRIC GENERATION FOR LONG-FORM DEEP RESEARCH

Anonymous authors
Paper under double-blind review

ABSTRACT

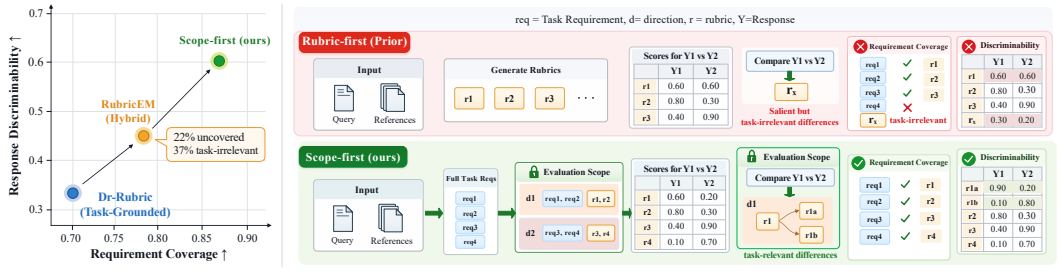
Long-form deep research (LFDR) tasks lack definitive answers for RL with verifiable rewards (RLVR), motivating rubric-based rewards for training deep research agents. Rubric effectiveness hinges on jointly covering full task requirements and distinguishing task-relevant quality differences among reports. Yet existing methods often miss implicit requirements or capture task-irrelevant response differences. We attribute both failures to an overlooked constraint: full task requirements are not explicitly elicited to constrain what rubrics should cover and which response differences they should capture. Accordingly, given the multifaceted nature of LFDR tasks, our key insight is to group requirements sharing an evaluation goal into directions that jointly form a task-defined evaluation scope, constraining both rubric generation and decomposition. We propose Scope2Rubric (S2R), a two-stage rubric construction framework. Stage I extracts full task requirements from query and reference analyses, organizes them into weighted directions, and reviews their coverage to form the evaluation scope. The scope constrains rubric generation within each direction, followed by individual, intra-direction, and cross-direction refinement to ensure the rubric set's necessity, completeness, and non-redundancy. Stage II identifies rubrics that fail to distinguish candidate responses and decomposes them by contrasting task-relevant differences within the fixed Stage I scope, preventing salient differences from inducing task-irrelevant rubrics. Extensive experiments show S2R achieves the best human-preference alignment over strong baselines. When used as rewards to train Qwen3-8B, it yields a superior 67.72 average score across five LFDR benchmarks. Further analyses confirm these gains stem from structured task-defined scope constraints.

1 INTRODUCTION

Large language models (LLMs) have shown impressive performance in long-form deep research (LFDR) (Shao et al., 2026), iteratively retrieving and reasoning to synthesize comprehensive reports (Li et al., 2026d; Zheng et al., 2025; Du et al., 2026). Reinforcement learning (RL) further enhances this long-horizon search capacity (Jin et al., 2025; Zheng et al., 2025; Shao et al., 2026). Since LFDR lacks definitive answers for RL with verifiable rewards (RLVR) (Lambert et al., 2024), rubric-based rewards provide checkable rubrics for policy optimization (Gunjal et al., 2026). The effectiveness of these rewards hinges on whether the rubrics comprehensively cover task requirements and distinguish task-relevant differences (Lv et al., 2026; Zhu et al., 2026; Shen et al., 2026).

Existing automated rubric construction uses task-grounded, response-grounded, or hybrid sources. Task-grounded methods derive rubrics from queries, evidence, or references (Lv et al., 2026; Zhu et al., 2026; Dhole & Agichtein, 2026), but often miss implicit requirements and remain too coarse to distinguish response quality. Response-grounded methods mine differences among candidate responses (Zhang et al., 2026b; Liu et al., 2026b; Kawabata & Sugawara, 2026), improving discriminability but potentially missing universally absent requirements and converting task-irrelevant differences into rubrics. Hybrid approaches combine task information with response differences to generate or refine rubrics (Liu et al., 2026c; Shen et al., 2026; Jia et al., 2026). As illustrated in Figure 1(a), this improves requirement coverage by 9% and discriminability by 12% over the single source baseline, yet leaves 22% of requirements uncovered and adds 37% of task-irrelevant rubrics.

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



(a) Coverage and Discriminability

(b) Rubric-first vs. Scope-first

Figure 1: **(a)** Hybrid methods improve coverage and discriminability yet leave omissions or task-irrelevant rubrics. **(b)** Rubric-first derives rubrics from task/response cues; S2R establishes the complete task scope.

Collectively, existing approaches' rubrics fail to jointly achieve complete task-requirement coverage and strong response discriminability. The shared limitation is the lack of a common task constraint: full task requirements are not explicitly elicited to constrain both what rubrics should cover for completeness and which response differences they may capture to improve discriminability. Empirically, applying this constraint improves coverage and discriminability by 8% and 13% over the hybrid baseline.This motivates a central question: how can full task requirements constrain rubric construction to achieve both complete task coverage and strong response discriminability?

Our key insight is to introduce a task-defined evaluation scope to make this constraint explicit in rubric construction: the scope first guides rubric generation to cover all task requirements, then restricts response-driven rubric decomposition to quality differences within those requirements. Given the multi-faceted evaluation needs of LFDR tasks (Zhu et al., 2026; Liu et al., 2024), we organize task requirements by evaluation goal into evaluation directions that capture their evaluation content and relations. These directions form an evaluation scope, a task-level representation that specifies what to evaluate and remains invariant across responses. Realizing this insight requires addressing two challenges: (1) how to construct a task-faithful scope that yields rubrics covering all task requirements, and (2) how to decompose rubrics using response quality differences within that scope.

To address above challenges, we propose Scope2Rubric (S2R), a novel two-stage rubric construction framework. For Challenge 1, Stage I clusters full requirements extracted from query and references by evaluation goals into weighted directions, each defining its evaluation content (semantics and boundaries) and structure (requirement relationships). A task coverage review ensures their necessity and global completeness to form the evaluation scope. Constrained within each defined direction, generated rubrics preserve directions' boundaries and weights, and then undergo individual, intra-direction, and cross-direction refinement to guarantee task necessity, completeness, and non-redundancy. For Challenge 2, Stage II identifies rubrics that fail to differentiate candidate responses. By contrasting these responses within the corresponding direction, it uncovers fine-grained quality differences and decomposes these rubrics into finer criteria without expanding their boundaries or altering their weights. Unlike prior "rubric-first" methods that derive rubrics directly from task and response cues, S2R establishes a task-level complete scope to constrain subsequent rubric generation and decomposition (scope-first; Figure 1(b)).

Extensive experiments across LFDR benchmarks demonstrate the effectiveness of Scope2Rubric. On human-preference alignment,Scope2Rubric achieves 76.39% and 53.15% accuracy on DeepResearchBench and ScholarQA-CS2, respectively; when used as rewards to train Qwen3-8B, it reaches an average score of 67.72 across five LFDR benchmarks, compared with 59.84 for the strongest evolving-rubric baseline. Further analyses show higher-quality evaluation directions and rubrics, while ablations attribute the gains to structured scope construction and scope-constrained rubric decomposition.Our contributions are threefold:

- We identify that the lack of holistic task constraints on response differentiation hinders the joint pursuit of a rubric's complete task coverage and high discriminability.
- We propose Scope2Rubric (S2R), a novel training-free framework featuring task-defined scopes formulated from complete requirements to constrain rubric generation with hierarchical reviews and subsequent response-driven decomposition for boosted discriminability.
- Extensive experiments show that under task-defined scope constraints, S2R generates higher-quality rubrics, achieves superior preference alignment, and outperforms strong baselines when used to train deep research agents.

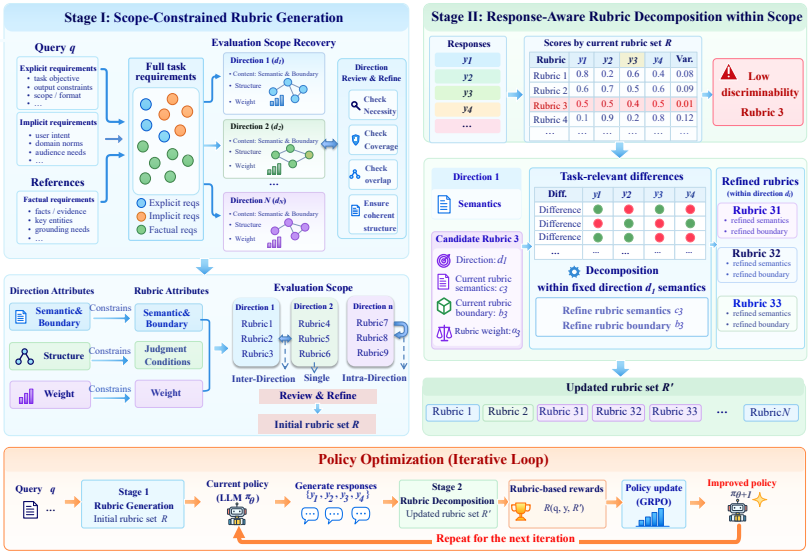


Figure 2: Overview of Scope2Rubric (S2R). Stage I recovers requirements and forms the weighted scope $\mathcal{S}_q$ and initial rubric set R. Stage II contrasts responses and decomposes low-discriminability rubrics within fixed directions, preserving boundaries and weights in R'.

2 Related Work

Rubrics provide fine-grained supervision for both LFDR evaluation and policy optimization. Existing generation methods are task-grounded (Lv et al., 2026; Dhole & Agichtein, 2026; Mei et al., 2026; Zhu et al., 2026), response-grounded (Zhang et al., 2026b; Liu et al., 2026b; Kawabata & Sugawara, 2026), or hybrid (Liu et al., 2026c; Jia et al., 2026; Shen et al., 2026; Li et al., 2026c). Task-grounded methods can miss requirements or yield coarse criteria, whereas response-grounded methods improve discriminability but may capture task-irrelevant differences or omit unrepresented requirements. For RL, static methods fix rubrics before training, while evolving approaches update or refine them from policy responses to maintain discriminative rewards (Shao et al., 2026; Li et al., 2026a; Mei et al., 2026; Piao et al., 2026; Zhang et al., 2026a). Yet unconstrained evolution can drift toward salient response differences rather than task-relevant quality. Scope2Rubric instead first establishes a task-defined evaluation scope from full task requirements, then uses current responses only to decompose insufficiently discriminative rubrics within that fixed scope.

3 Methodology

Existing methods may miss implicit requirements or turn salient response differences into task-irrelevant criteria, compromising coverage and discriminability. As shown in Figure 2, S2R addresses both in two stages: Stage I recovers the task-defined scope and constructs the initial rubric set; Stage II decomposes low-discriminability rubrics using response differences within this fixed scope. Section 3.1 defines the notation before detailing both stages and rubric-based rewards.

3.1 Problem Formulation

Given an LFDR query q, let $\mathcal{U}_q = \{u_j\}_{j=1}^{n_q}$ denote its full task requirements, i.e., all query-specific conditions that should affect report-quality evaluation. Requirements with a common evaluation goal form an evaluation direction d_i. Each direction contains its requirements $\mathcal{U}_i \subseteq \mathcal{U}_q$, evaluation content e_i, evaluation structure ϕ_i, and weight w_i; all directions form the evaluation scope $\mathcal{S}_q$:

$$e_i = (c_i, b_i), \qquad d_i = (\mathcal{U}_i, e_i, \phi_i, w_i), \qquad \mathcal{S}_q = \{d_i\}_{i=1}^{m}. \tag{1}$$

Here, semantics c_i specifies what is evaluated and its quality meaning, while boundaries b_i define included and excluded content. ϕ_i captures how requirement relations jointly determine the evaluation goal, and w_i denotes the direction's relative task importance.

Each direction d_i is translated into a set of checkable rubrics, denoted by $\mathcal{R}_{q,i}$.

$$e_r = (c_r, b_r), \qquad r = (e_r, a_r, \alpha_r), \qquad \mathcal{R}_q = \bigcup_{i=1}^{m} \mathcal{R}_{q,i}. \tag{2}$$

Here, c_r specifies the particular quality aspect evaluated within c_i, while b_r defines its applicable content range. The judgment conditions a_r specify the conditions and evidence for judgments, and α_r denotes the rubric's relative importance within its direction.

3.2 Scope-Constrained Rubric Generation

Task-Grounded Scope Recovery. Direct rubric generation from task cues may omit necessary requirements or introduce task-irrelevant content. Stage I therefore first recovers the full task requirements. Explicit constraints are extracted from the query, while implicit requirements necessary for task completion are discovered and verified across research stages using task context such as user intent and domain norms (Zhou et al., 2026a). Reference-answer evidence further supplements factual requirements requiring retrieval, with cross-answer agreement and task-necessity checks filtering reference-specific but task-irrelevant content.The recovered requirements are grouped by evaluation goal into directions. Since initial directions may be inaccurate, incomplete, or redundant, review checks each direction for correctness and necessity, all directions for task-requirement coverage and redundancy, and assigns w_i by task importance and omission impact. Iterative revision and normalization yield $\mathcal{S}_q$ with $\sum_{i=1}^{m} w_i = 1$.

Scope-Constrained Rubric Generation and Hierarchical Refinement. Even with the full task requirements recovered, generated rubrics may omit required content or duplicate evaluation. Stage I therefore generates rubrics under $\mathcal{S}_q$ and hierarchically refines them for complete and non-redundant coverage. For each d_i, e_i constrains e_r within its semantics and boundaries, while ϕ_i constrains a_r so that requirement relations are reflected in concrete evaluations.

Each α_r is assigned according to its contribution to required content and omission impact, then normalized within d_i. Since w_i remains fixed,

$$\mathcal{R}^0_{q,i} = \mathrm{Generate}(q \mid d_i), \qquad \sum_{r \in \mathcal{R}^0_{q,i}} \alpha_r = 1 \implies \sum_{r \in \mathcal{R}^0_{q,i}} w_i \alpha_r = w_i. \tag{3}$$

Thus, a direction's total contribution remains w_i regardless of rubric count.The generated rubrics are then refined at three levels. Single-rubric refinement checks direction consistency, task necessity, atomicity, and judgeability. Intra-direction refinement checks coverage, semantic or behavioral redundancy, and weight appropriateness. Inter-direction refinement removes duplicate evaluation across directions while retaining task-required content unique to each. Revision continues until $\mathcal{R}^0_q = \bigcup_i \mathcal{R}^0_{q,i}$ correctly covers the full task requirements.

3.3 Response-Aware Rubric Decomposition within Scope

Complete task-requirement coverage does not ensure sufficient discriminability, as some rubrics may remain too coarse to capture task-relevant differences among current responses. Stage II therefore scores the current responses with $\mathcal{R}^0_q$ and decomposes low-discriminability rubrics only when unresolved task-relevant differences exist within their evaluation directions.

Let $\mathcal{Y}_t(q) = \{y_{t,k}\}_{k=1}^{K}$ denote the current responses. For $r \in \mathcal{R}^0_{q,i}$ under d_i, let Δ^t_r denote quality differences in $\mathcal{Y}_t(q)$ that fall within r's evaluation content $e_r = (c_r, b_r)$, remain consistent with d_i, but are not distinguished by a_r. Rubrics selected for decomposition satisfy

$$\mathcal{A}_t(q) = \left\{ r \in \mathcal{R}^0_q \;\middle|\; \mathrm{LowDisc}(r, \mathcal{Y}_t(q)) \wedge \Delta^t_r \neq \emptyset \right\}. \tag{4}$$

LowDisc only identifies low-discriminability candidates. If $\Delta^t_r = \emptyset$, including uniform satisfaction or failure without further task-relevant differences, r is retained.

For each $r \in \mathcal{A}_t(q) \cap \mathcal{R}^0_{q,i}$, Stage II decomposes r according to Δ^t_r under d_i while preserving its within-direction weight:

$$\mathcal{C}^t_r = \mathrm{Decompose}\big(r, \Delta^t_r \mid d_i\big), \qquad \sum_{r' \in \mathcal{C}^t_r} \alpha^t_{r'} = \alpha_r. \tag{5}$$

Because visible response differences may omit necessary shared content or introduce redundancy, $\mathcal{C}_r^t$ is reviewed under d_i to collectively preserve r's necessary evaluation content, judgment conditions, and boundary while removing redundancy. Only a validated decomposition replaces r.

Throughout Stage II, $\mathcal{S}_q$ remains fixed. Response differences determine only how low-discriminability rubrics are decomposed within existing task requirements, without changing evaluation goals or direction weights.

3.4 Rubric-Based Reward for Policy Optimization

Before training, Stage I constructs $\mathcal{S}_q$ and $\mathcal{R}_q^0$ for each query q. At update step t, π_{θ_t} samples $\mathcal{Y}_t(q) = \{y_{t,k}\}_{k=1}^K$ with $y_{t,k} \sim \pi_{\theta_t}(\cdot \mid q)$. Scores under $\mathcal{R}_q^0$ determine $\mathcal{A}_t(q)$, after which Stage II produces the decompositions in Eq. equation 5. The current rubric set is

$$\mathcal{R}_q^t = \big(\mathcal{R}_q^0 \setminus \mathcal{A}_t(q)\big) \cup \bigcup_{r \in \mathcal{A}_t(q)} \mathcal{C}_r^t. \tag{6}$$

Scores for unchanged rubrics are reused, and only newly decomposed rubrics are evaluated on the same responses. If $\mathcal{A}_t(q) = \emptyset$, then $\mathcal{R}_q^t = \mathcal{R}_q^0$.

Let $s_r^t(q, y)$ denote the resulting evaluator score and $\mathcal{R}_{q,i}^t$ the current rubrics under d_i. Equation equation 5 preserves $\sum_{r \in \mathcal{R}_{q,i}^t} \alpha_r^t = 1$, so decomposition only redistributes the original within-direction weight while w_i remains fixed. The reward used by standard GRPO (Shao et al., 2024) is

$$S_i^t(q, y) = \sum_{r \in \mathcal{R}_{q,i}^t} \alpha_r^t s_r^t(q, y), \qquad R_t(q, y) = \sum_{i=1}^m w_i S_i^t(q, y) = \sum_{i=1}^m w_i \sum_{r \in \mathcal{R}_{q,i}^t} \alpha_r^t s_r^t(q, y). \tag{7}$$

Each $y \in \mathcal{Y}_t(q)$ receives $R_t(q, y)$ for the subsequent GRPO update. Thus, Stage I preserves task-requirement coverage through the fixed scope, while Stage II adapts rubric granularity to provide task-relevant reward signals for the current responses.

4 Experiments

4.1 Experimental Setup

For human preference alignment, comparisons cover rubric-free Wang et al. 2024 and rubric-guided reward models Dhole & Agichtein 2026; Mei et al. 2026; Li et al. 2026a on DeepResearch Bench (DRB) Du et al. 2026 and ScholarQA-CS2 (SQAv2) Bragg et al. 2026, using accuracy and paired Cohen's d Lv et al. 2026. For DeepResearch agent evaluation, we evaluate rubric-trained Qwen3-8B-SFT on HealthBench (HB) Arora et al. 2025 for healthcare, ResearchQA (RQA) Yifei et al. 2026 and SQAv2 for literature synthesis, and DRB and DeepResearch Bench II (DRB2) Li et al. 2026b for research. Baselines include proprietary systems OpenAI 2025; Perplexity AI 2025, rubric-free agents Li et al. 2026d, and static Zhu et al. 2026 or evolving rubric RL Shao et al. 2026.Comprehensive training and evaluation details appear in Appendix H.

4.2 Main Results

Human Preference Alignment Evaluation. To examine whether a task-defined scope improves human-preference alignment, Table 1 compares Scope2Rubric with rubric-free, task-grounded, response-grounded, and hybrid baselines on DR Bench and SQAv2. Scope2Rubric (Gemini3) achieves 76.39%/53.15% accuracy and 0.69/0.35 paired d, surpassing the strongest baselines by 2.23/2.55 points and 0.03/0.09; the average accuracy is 64.77%, 3.33 points above RubricEM. Stage I recovers task content, boundaries, and priorities, while Stage II decomposes coarse rubrics only within this scope, avoiding missing requirements and task-irrelevant response differences.

Table 1: Main results on human preference alignment for LFDR tasks, reporting pairwise accuracy and Cohen's d. Gemini-2.5-Pro serves as the common judge for all rubric-guided methods. Best and second-best results are marked in bold and underlined. Full results appear in Appendix I.1.

Model	Rubric Construction	DR Bench		SQAv2		Avg.	
		Acc.↑	d ↑	Acc.↑	d ↑	Acc.↑	d ↑
Rubric-Free Reward Models							
GPT-5.4	N/A	64.09	–	45.05	–	54.57	–
Gemini-3.1-Pro	N/A	67.80	–	49.07	–	58.44	–
ArmoRM-Llama3-8B	N/A	44.63	-0.11	44.99	-0.19	44.81	-0.15
Rubric-Guided Reward Models							
Tournament-GRPO	Task-Grounded	72.48	0.48	49.66	0.21	61.07	0.34
QUEST	Task-Grounded	71.14	0.58	50.60	0.26	60.87	0.42
RubricRAG	Task-Grounded	44.63	0.19	40.73	0.18	42.68	0.18
DR-Rubric	Task-Grounded	72.35	0.53	42.84	0.24	57.60	0.38
DeepRubric	Task-Grounded	70.28	0.49	45.03	0.21	57.66	0.35
HOTE	Response-Grounded	67.79	0.37	44.73	0.12	56.26	0.25
RubricEM	Hybrid	74.16	<u>0.66</u>	48.72	0.21	61.44	0.44
Ours							
Scope2Rubric (GPT5)	Hybrid	<u>75.10</u>	0.64	<u>52.84</u>	<u>0.32</u>	<u>63.97</u>	<u>0.48</u>
Scope2Rubric (Gemini3)	Hybrid	**76.39**	**0.69**	**53.15**	**0.35**	**64.77**	**0.52**

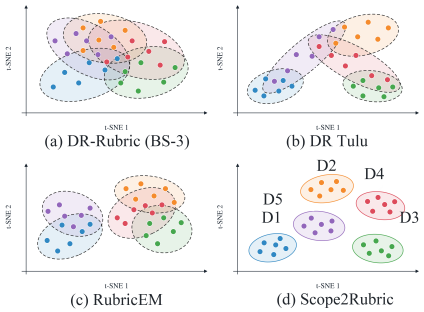
Policy Optimization with Rubric as Rewards. To answer whether scope-constrained rubrics can turn task-relevant quality distinctions into effective reward signals for policy optimization, Table 2 compares downstream performance across five LFDR benchmarks. Scope2Rubric achieves the best average score of 67.7, outperforming the strongest rubric-free, static-rubric, and evolving-rubric baselines by 4.8, 5.6, and 7.9 points, respectively. Gains are largest on DR Bench2 (+9.9) and HealthBench (+10.8), where fine-grained task requirements make complete task coverage especially important, supporting Stage I's task-defined evaluation scope and hierarchical refinement. Consistent gains on ResearchQA (+3.6) and SQAv2 (+2.1) further suggest that response-aware rubric decomposition within the fixed scope improves task-relevant discriminability without expanding direction boundaries or altering direction weights.

4.3 Ablation Studies

The contributions of the Scope2Rubric's three key mechanisms are examined through w/o Structured Scope, w/o Scope Constraint, and w/o Fixed Scope in Table 3. Removing structured scope reduces preference and downstream performance by 2.38/2.20 points, showing the benefit of organizing requirements by shared evaluation goals. Removing scope constraints causes the largest drops of 4.11/3.60 points, as unconstrained response differences can introduce task-irrelevant rubric. Evolving the scope also loses 3.26/3.00 points, showing that adaptation benefits from preserving task requirements while refining rubric granularity.

Table 3: Scope2Rubric ablations across four variants. Gemini3 generates rubrics; preference is measured before RL; downstream scores use Qwen3-8B-SFT after RL.

Variant	Preference Align		Downstream	
	DRB	SQAv2	DRB2	RQA
Scope2Rubric	**76.39**	**53.15**	**48.40**	**80.50**
w/o Structured Scope	73.88	50.91	46.10	78.40
w/o Scope Constraint	71.96	49.37	44.80	76.90
w/o Fixed Scope	72.85	50.18	45.40	77.50

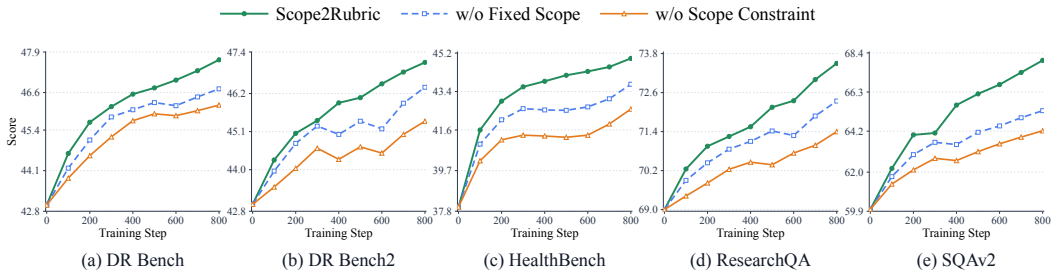


(a) DR-Rubric (BS-3) (b) DR Tulu (c) RubricEM (d) Scope2Rubric

Figure 3: Rubric set semantic distributions.

Table 2: Downstream performance of representative LFDR systems. * denotes scores reported in prior work; † denotes 100-sample evaluation. Dashes indicate scores not reported in the cited source. Avg. averages available scores. Bold and underlined values identify the best and second-best open-model results. DualGraph uses GPT-5 for DR Bench and GPT-4.1 for DR Bench2. The full comparison appears in Appendix I.2 (Table 8).

Model	Backbone	DRB	DRB2	HB	RQA	SQAv2	Avg.
Proprietary DeepResearch							
Claude-Sonnet Search	Claude Sonnet	34.5*	–	–	64.3*	–	49.4
Perplexity Deep Research	Proprietary	42.3*	38.6*	–	75.3*	67.3*	55.9
Gemini Deep Research	Proprietary	48.8*	42.0*	–	68.5*	–	53.1
OpenAI Deep Research	o3/o4	46.9*	45.4*	53.8*,†	79.2*,†	79.6*	61.0
Gemini 3 Pro + Search	Gemini 3 Pro	46.3*	34.6	38.0*	74.3*	69.8*	52.6
GPT-5 + Search	GPT-5	50.7*	–	59.5*,†	78.2*,†	74.8*	65.8
Rubric-Free DeepResearch							
FS-Researcher	GPT-5	52.8*	40.7	59.3	79.9	79.8	62.5
DualGraph	GPT-5 / GPT-4.1	<u>53.1</u>*	41.5*	60.0	78.8	81.1	62.9
WebWeaver	GPT-4.1	45.2*	36.6*	53.3	76.5	78.8	58.1
RL with Static Rubric Rewards for DeepResearch							
DeepResearch Gen.	Tongyi-30B-A3B	49.3*	40.4	47.9	78.5	85.4	60.3
Tournament-GRPO	Qwen2.5-7B	50.2*	38.3	53.6	79.4	86.1	61.5
DeepRubric	Qwen3-8B	43.6*	33.5	45.1	75.2*	86.0*	56.7
Quest-35B	Qwen3.5-35B	48.2*	42.6	57.2	77.1	85.6	62.1
RL with Evolving Rubric Rewards for DeepResearch							
DR Tulu	Qwen3-8B	45.4*	37.0	52.8*	75.7*	88.3*	59.8
DR-Rubric (BS-3)	Qwen3-8B	39.5*	31.6	46.9	72.4*	84.1	54.9
RubricEM	Qwen3-8B	47.8*	38.5	49.3*	74.5*	85.8	59.2
ARISE-RL	Qwen3-8B	46.1	35.1	52.7	74.8	82.4	58.2
HOTE	Qwen3-8B	45.9*	32.4	54.4*	76.9*	80.5	58.0
Ours							
Qwen3-8B-SFT	Qwen3-8B	38.4	29.6	35.6	59.3	60.9	44.8
Scope2Rubric (GPT5)	Qwen3-8B	52.5	<u>47.2</u>	<u>62.1</u>	**82.8**	<u>89.7</u>	<u>66.9</u>
Scope2Rubric (Gemini3)	Qwen3-8B	**54.1**	**48.4**	**65.2**	<u>80.5</u>	**90.4**	**67.7**



(a) DR Bench (b) DR Bench2 (c) HealthBench (d) ResearchQA (e) SQAv2

Figure 4: Ablation training dynamics across five LFDR benchmarks. Each panel shows downstream score against training step for Scope2Rubric, w/o Fixed Scope, and w/o Scope Constraint.

Figure 4 further shows how the two scope mechanisms affect policy optimization as responses evolve. The variants perform similarly early in training, but Scope2Rubric gradually pulls ahead of both ablations across all five benchmarks. The widening gap to w/o Fixed Scope shows the benefit of fixing the task-defined scope: it prevents response-dependent scope drift and preserves requirements temporarily absent from rollouts. The larger gap to w/o Scope Constraint reflects scope-constrained decomposition, which increases rubric granularity only for task-relevant response differences and maintaining stable, task-relevant supervision throughout training.

4.4 Empirical Analysis

To examine whether task-defined scope resolves the coverage–discriminability trade-off, this section first evaluates recovered scope quality, then analyzes generated rubrics through expert alignment and intra-/inter-direction set quality. It further examines training dynamics and efficiency, followed by cross-domain generalization to assess whether the resulting policy gains transfer beyond LFDR task.

Evaluation Scope Quality Analysis. Incomplete requirement coverage and overlapping evaluation directions motivate jointly examining scope quality and requirement organization in Figures 5 and 6. Scope2Rubric improves correctness and coverage over DR-Rubric by 4.22–6.48 and 2.78–4.65 points, raises necessity by 0.42–0.50, and reduces redundancy by 5.18–6.35 points. Importantly, higher coverage does not result from indiscriminate scope expansion: the requirement–direction map shows requirements concentrated in primary directions with limited secondary overlap. This pattern reflects scope-first recovery, which groups complete task requirements by shared evaluation goals, establishing distinct direction ownership while retaining legitimate dependencies. Consequently, broader task coverage is achieved without repeated judgments, while the 10.67–14.35-point weight gains show that recovered directions preserve meaningful differences in task importance.

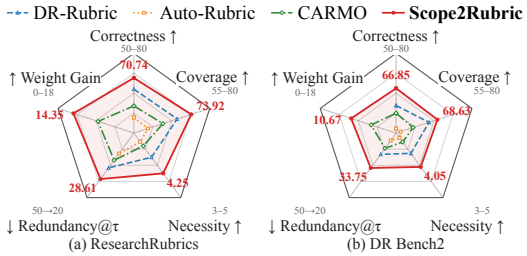


Figure 5: Scope quality: ResearchRubrics and DRB2; Scope2Rubric in red.

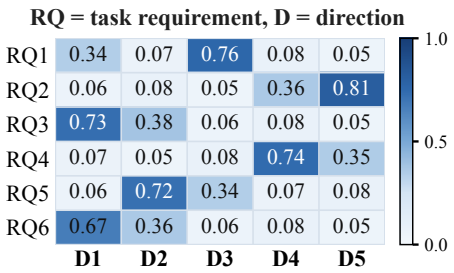


Figure 6: Requirement coverage across directions.

Generated Rubrics Aligned with Expert-Written Rubrics. Quantitatively, Scope2Rubric raises semantic F1 from 0.43/0.47 to 0.51/0.53 over RubricEM, while consistently improving ρ, τ_b, Top-1, and pairwise accuracy. Higher recall reflects recovery of missing task requirements, whereas improved precision and ranking agreement indicate that direction boundaries suppress irrelevant criteria while within-scope decomposition exposes finer task-relevant differences. Qualitatively, Figure 3 shows the resulting semantic structure: Scope2Rubric forms more compact, clearly separated direction-specific rubric clusters with substantially less cross-direction overlap than competing methods. This organization reflects scope-first construction, which assigns distinct evaluation content to directions before refining rubric granularity within their fixed boundaries.

Table 4: Generated rubrics are compared with expert-written rubrics for semantic and behavioral alignment. Behavioral metrics are Spearman rank correlation (ρ), Kendall's tau-b (τ_b), top-1 consistency, and pairwise accuracy. Bold and underlined scores mark the best and second-best results within each dataset.

Model	Semantic Alignment			Behavior Alignment			
	Precision	Recall	F1	ρ	τ_b	Top-1	Pairwise
HealthBench-Consensus							
Tournament-GRPO	0.32	0.48	0.38	0.35	0.26	0.27	0.52
RubricEM	<u>0.36</u>	<u>0.52</u>	<u>0.43</u>	<u>0.39</u>	<u>0.30</u>	<u>0.29</u>	<u>0.55</u>
Scope2Rubric (Gemini-3)	**0.45**	**0.59**	**0.51**	**0.44**	**0.36**	**0.35**	**0.60**
ResearchRubrics							
Tournament-GRPO	0.39	0.50	0.44	0.41	0.34	0.30	0.58
RubricEM	<u>0.42</u>	<u>0.54</u>	<u>0.47</u>	<u>0.45</u>	<u>0.38</u>	<u>0.33</u>	<u>0.61</u>
Scope2Rubric (Gemini-3)	**0.47**	**0.62**	**0.53**	**0.49**	**0.40**	**0.38**	**0.65**

Intra- and Inter-Direction Rubric Set Quality. Across the two refinement levels in Figure 7, Scope2Rubric raises intra-direction coverage to 80.93 while reducing semantic and behavioral redundancy to 19.64 and 0.28. Relative to DR-Rubric, this is a 6.88-point coverage gain with 8.21/0.11-point redundancy reductions. These coupled gains reflect scope-first organization followed by scope-constrained refinement: explicit direction boundaries allow missing judgments to be added within each direction without duplicating neighboring criteria. The 5.48-point weight utility, versus 2.88 for CARMO, further shows that within-direction weighting preserves differences in rubric importance. Inter-direction behavioral and semantic redundancy fall to 0.20/10.73 from

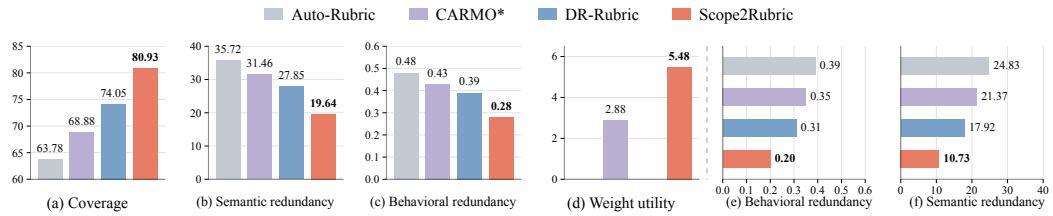


Figure 7: Intra- and inter-direction rubric-set quality across four methods. Panels (a)–(d) report intra-direction coverage, semantic and behavioral redundancy, and weight utility; panels (e)–(f) report inter-direction behavioral and semantic redundancy.

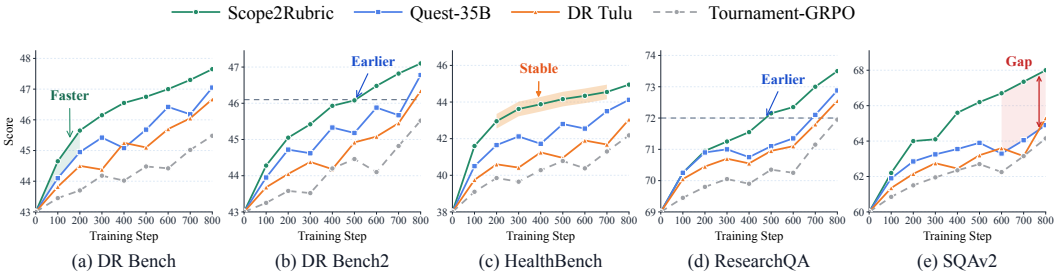


Figure 8: Training dynamics and step efficiency of Scope2Rubric across five LFDR benchmarks.

DR-Rubric's 0.31/17.92, showing that fixed direction boundaries prevent repeated evaluation of the same content or response behavior across directions.

Training Dynamics and Efficiency Analysis. Across the five benchmarks in Figure 8, Scope2Rubric learns faster, reaches comparable performance earlier on DR Bench2 and ResearchQA, maintains smoother gains on HealthBench, and expands its advantage at later steps on SQAv2. This pattern reflects the interaction between a fixed task-defined scope and response-aware decomposition within scope. The fixed scope keeps reward targets aligned with the same task requirements as policy outputs evolve, avoiding supervision drift; decomposition then increases rubric granularity only where current responses remain insufficiently distinguished. Their combination therefore supplies finer yet stable task-relevant rewards, explaining both earlier threshold attainment and sustained late-stage gains rather than only a higher final score.

Cross-Domain Generalization. Scope2Rubric reaches the highest average of 75.5, improving over RubricEM by 1.6/6.7/2.1 points on 2Wiki, WebWalker, and DSQA. These gains suggest scope-first rewards transfer well when tasks require multiple task-relevant distinctions: the fixed scope preserves required objectives, while within-scope decomposition sharpens partial progress. SimpleQA drops 2.5 points, likely because single-fact correctness gains little from finer rubrics and favors exact matching.

Table 5: Short-form search performance. SQA denotes SimpleQA and DSQA denotes DeepSearchQA; * denotes scores reported in prior work (Li et al., 2026a).

Model	SQA	2Wiki	WebWalker	DSQA	Avg.
DR Tulu-8B	80.1*	68.0*	39.1*	8.3*	49.0*
RubricEM	**92.3***	78.8*	70.0*	53.0*	73.5*
Tournament-GRPO	85.6	75.2	74.2	50.4	71.4
Ours	89.8	**80.4**	**76.7**	**55.1**	**75.5**

5 Conclusion

In this paper, we present Scope2Rubric, a scope-first framework for constructing query-specific rubrics for LFDR. Stage I recovers full task requirements into weighted evaluation directions, preventing omissions and fixing what should be evaluated before rubric generation. Stage II then decomposes low-discriminability rubrics only within these fixed boundaries, improving response discrimination without admitting task-irrelevant differences or shifting task priorities. This design improves preference alignment, expert consistency, and downstream policy optimization. However, its effectiveness still depends on accurate scope recovery, reference quality, and LLM-based review and judging, which may propagate omissions or scoring errors.

AI USE STATEMENT

Generative AI assisted with manuscript drafting and editing, literature organization, and LaTeX preparation. Appendix A describes these uses. The authors are responsible for the final manuscript.

REFERENCES

Rahul K. Arora, Jason Wei, Rebecca Soskin Hicks, et al. HealthBench: Evaluating large language models towards improved human health, 2025.

Jonathan Bragg, Mike D'Arcy, Nishant Balepur, et al. AstaBench: Rigorous benchmarking of ai agents with a scientific research suite. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=M7TNf5J26u.

Xiusi Chen, Gaotang Li, Ziqi Wang, et al. RM-R1: Reward modeling as reasoning. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=1ZqJ6jj75q.

Yifan Chen, Haitao Li, Qingyao Ai, Fengbin Zhu, Tat-Seng Chua, Min Zhang, and Yiqun Liu. GenRubric: Self-Evolving Rubric Generation for Scalable LLM Evaluation, 2026b. URL https://arxiv.org/abs/2608.29856.

Dave Citron. Try Deep Research and our new experimental model in Gemini, your AI assistant, December 2024. URL https://blog.google/products-and-platforms/products/gemini/google-gemini-deep-research/. Google product announcement, December 11, 2024.

Kaustubh D. Dhole and Eugene Agichtein. RubricRAG: Towards interpretable and reliable llm evaluation via domain knowledge retrieval for rubric generation. In *Proceedings of the 49th International ACM SIGIR Conference on Research and Development in Information Retrieval*, pp. 3681–3687, 2026. doi: 10.1145/3805712.3809940. URL https://doi.org/10.1145/3805712.3809940.

Mingxuan Du, Benfeng Xu, Chiwei Zhu, et al. DeepResearch Bench: A comprehensive benchmark for deep research agents. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=hQ0K2Hhq7H.

Anisha Gunjal, Anthony Wang, Elaine Lau, et al. Rubrics as rewards: Reinforcement learning beyond verifiable domains. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=c1bTcrDmt4.

Nikita Gupta, Riju Chatterjee, Lukas Haas, Connie Tao, Andrew Wang, Chang Liu, Hidekazu Oiwa, Elena Gribovskaya, Jan Ackermann, John Blitzer, Sasha Goldshtein, and Dipanjan Das. DeepSearchQA: Bridging the comprehensiveness gap for deep research agents, 2026. URL https://arxiv.org/abs/2601.20975.

Taneesh Gupta, Shivam Shandilya, Xuchao Zhang, et al. CARMO: Dynamic criteria generation for context aware reward modelling. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 2202–2261, Vienna, Austria, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.findings-acl.114. URL https://aclanthology.org/2025.findings-acl.114/.

Xanh Ho, Anh-Khoa Duong Nguyen, Saku Sugawara, and Akiko Aizawa. Constructing a multi-hop QA dataset for comprehensive evaluation of reasoning steps. In *Proceedings of the 28th International Conference on Computational Linguistics*, pp. 6609–6625. International Committee on Computational Linguistics, 2020. doi: 10.18653/v1/2020.coling-main.580. URL https://aclanthology.org/2020.coling-main.580/.

Ruipeng Jia, Yunyi Yang, Yuxin Wu, et al. Open rubric system: Scaling reinforcement learning with pairwise adaptive rubric, 2026. URL https://arxiv.org/abs/2602.14069.

Bowen Jin, Hansi Zeng, Zhenrui Yue, et al. Search-R1: Training llms to reason and leverage search engines with reinforcement learning. In *Proceedings of the Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=Rwhi91ideu.

Akira Kawabata and Saku Sugawara. C2: Scalable rubric-augmented reward modeling from binary preferences. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 11397–11424, San Diego, California, United States, 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.523.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V. Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, Yuling Gu, Saumya Malik, Victoria Graf, Jena D. Hwang, Jiangjiang Yang, Ronan Le Bras, Oyvind Tafjord, Chris Wilhelm, Luca Soldaini, Noah A. Smith, Yizhong Wang, Pradeep Dasigi, and Hannaneh Hajishirzi. Tulu 3: Pushing frontiers in open language model post-training, 2024. URL https://arxiv.org/abs/2411.15124.

Gaotang Li, Bhavana Dalvi Mishra, Zifeng Wang, Jun Yan, Yanfei Chen, Chun-Liang Li, Long T. Le, Rujun Han, George Lee, Hanghang Tong, Chen-Yu Lee, and Tomas Pfister. RubricEM: Meta-RL with rubric-guided policy decomposition beyond verifiable rewards, 2026a. URL https://arxiv.org/abs/2605.10899.

Ruizhe Li, Mingxuan Du, Benfeng Xu, et al. DeepResearch Bench II: Diagnosing deep research agents via rubrics from expert report, 2026b.

Sunzhu Li, Jiale Zhao, Huimin Ren, et al. RubricHub: A comprehensive and highly discriminative rubric dataset via automated coarse-to-fine generation. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 31320–31344, San Diego, California, United States, 2026c. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.1445.

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. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, pp. 120091–120131. Curran Associates, Inc., 2025. doi: 10.52202/085713-4011. URL https://proceedings.neurips.cc/paper_files/paper/2025/hash/ae03bdef276132fae089692445725635-Abstract-Conference.html.

Zijian Li, Xin Guan, Bo Zhang, et al. WebWeaver: Structuring web-scale evidence with dynamic outlines for open-ended deep research. In *The Fourteenth International Conference on Learning Representations*, 2026d. URL https://openreview.net/forum?id=MtNCJjlrKt.

Chris Yuhao Liu, Liang Zeng, Yuzhen Xiao, et al. Skywork-Reward-V2: Scaling preference data curation via human-ai synergy. In *The Fourteenth International Conference on Learning Representations*, 2026a. URL https://openreview.net/forum?id=ofgxkMLqic.

Dengcan Liu, Fengkai Yang, Xiaohan Wang, et al. CDRRM: Contrast-driven rubric generation for reliable and interpretable reward modeling, 2026b. URL https://arxiv.org/abs/2603.08035.

Tianci Liu, Ran Xu, Tony Yu, et al. OpenRubrics: Towards scalable synthetic rubric generation for reward modeling and llm alignment. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 17417–17437, San Diego, California, United States, 2026c. Association for Computational Linguistics. doi: 10.18653/v1/2026.acl-long.791. URL https://aclanthology.org/2026.acl-long.791/.

Yuxuan Liu, Tianchi Yang, Shaohan Huang, et al. HD-Eval: Aligning large language model evaluators through hierarchical criteria decomposition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 7630–7661, Bangkok, Thailand, 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.413. URL https://aclanthology.org/2024.acl-long.413/.

Zijun Liu, Peiyi Wang, Runxin Xu, Shirong Ma, Chong Ruan, Peng Li, Yang Liu, and Yu Wu. Inference-time scaling for generalist reward modeling, 2025. URL https://arxiv.org/abs/2504.02495.

Changze Lv, Jie Zhou, Wentao Zhao, Jingwen Xu, Shihan Dou, Zisu Huang, Muzhao Tian, Xiaohua Wang, Zhengkang Guo, Yang Liu, Pluto Zhou, Tao Gui, Le Tian, Xiao Zhou, Xiaoqing Zheng, Xuanjing Huang, and Jie Zhou. Learning query-specific rubrics from human preferences for DeepResearch report generation, 2026. URL https://arxiv.org/abs/2602.03619.

Wangyi Mei, Zhouhong Gu, Zhenhan Bai, et al. Deep research as rubric for reinforcement learning, 2026.

OpenAI. Introducing deep research, February 2025. URL https://openai.com/index/introducing-deep-research/. Official product announcement, February 2, 2025.

Perplexity AI. Introducing Perplexity Deep Research, February 2025. URL https://www.perplexity.ai/hub/blog/introducing-perplexity-deep-research. Official product announcement, February 14, 2025.

Hongming Piao, Chi Liu, Mengzhuo Chen, Yan Shu, Xidong Wang, Derek Li, Ying Wei, and Bryan Dai. Hybrid open-ended tri-evolution makes better deep researcher, 2026. URL https://arxiv.org/abs/2606.13710v2.

Zhengyang Qi, Charles Dickens, Derek Pham, Amanda Dsouza, Armin Parchami, Frederic Sala, and Paroma Varma. RIFT: A RubrIc Failure Mode Taxonomy and Automated Diagnostics, 2026. URL https://arxiv.org/abs/2604.01375.

Swastik Roy, Rajkumar Pujari, Tharindu Kumarage, Charith Peris, Rahul Gupta, Anna Rumshisky, Pradeep Natarajan, and Venkatesh Saligrama. PReMISE: Policy Rubrics as Measurement Specifications for LLM Judges, 2026. URL https://arxiv.org/abs/2605.30803.

Rulin Shao, Akari Asai, Shannon Zejiang Shen, Hamish Ivison, Varsha Kishore, Jingming Zhuo, Xinran Zhao, Molly Park, Samuel G. Finlayson, David Sontag, Tyler Murray, Sewon Min, Pradeep Dasigi, Luca Soldaini, Faeze Brahman, Wen tau Yih, Tongshuang Wu, Luke Zettlemoyer, Yoon Kim, Hannaneh Hajishirzi, and Pang Wei Koh. DR Tulu: Reinforcement learning with evolving rubrics for deep research. In *Proceedings of the 43rd International Conference on Machine Learning*, volume 306, 2026. URL https://openreview.net/forum?id=97NEP1pyS3.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models, 2024. URL https://arxiv.org/abs/2402.03300v3.

Manasi Sharma, Chen Bo Calvin Zhang, Chaithanya Bandi, et al. ResearchRubrics: A benchmark of prompts and rubrics for evaluating deep research agents. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=ErnvfmSX0P.

William F. Shen, Xinchi Qiu, Chenxi Whitehouse, Lisa Alazraki, Shashwat Goel, Francesco Barbieri, Timon Willi, Akhil Mathur, and Ilias Leontiadis. Rethinking rubric generation for improving LLM judge and reward modeling for open-ended tasks, 2026. URL https://arxiv.org/abs/2602.05125.

Divyansh Singh, Reza Davari, and Afra Mashhadi. RADAR: Rubric-Aware Dependency and Redundancy Analysis for LLM-as-Judge Evaluation, 2026. URL https://arxiv.org/abs/2608.01810.

Tongyi DeepResearch Team, 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, Minpeng Liao, 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, 2025. URL https://arxiv.org/abs/2510.24701.

Tongyi DeepResearch Team, 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, Minpeng Liao, 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, 2026. URL https://arxiv.org/abs/2510.24701.

Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. Interpretable preferences via multi-objective reward modeling and mixture-of-experts. In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 10582–10592, Miami, Florida, USA, November 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.findings-emnlp.620. URL https://aclanthology.org/2024.findings-emnlp.620/.

Jason Wei, Nguyen Karina, Hyung Won Chung, Yunxin Joy Jiao, Spencer Papay, Amelia Glaese, John Schulman, and William Fedus. Measuring short-form factuality in large language models, 2024. URL https://arxiv.org/abs/2411.04368.

Jialong Wu, Wenbiao Yin, Yong Jiang, Zhenglin Wang, Zekun Xi, Runnan Fang, Linhai Zhang, Yulan He, Deyu Zhou, Pengjun Xie, and Fei Huang. WebWalker: Benchmarking LLMs in web traversal. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10290–10305. Association for Computational Linguistics, 2025. doi: 10.18653/v1/2025.acl-long.508. URL https://aclanthology.org/2025.acl-long.508/.

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 From Implicit Weights to Explicit Rubrics for Reward Modeling, 2026. URL https://arxiv.org/abs/2510.17314.

Ran Xu, Tianci Liu, Zihan Dong, Tony Yu, Ilgee Hong, Carl Yang, Linjun Zhang, Tao Zhao, and Haoyu Wang. Alternating reinforcement learning for rubric-based reward modeling in non-verifiable llm post-training. In *Proceedings of the 43rd International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=3SDg8dtkS8.

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. *Transactions of the Association for Computational Linguistics*, 14:1344–1368, 2026. doi: 10.1162/tacl.a.732.

Fanrui Zhang, Ruixue Ding, Qiang Zhang, Xi Chen, Boli Chen, Shihang Wang, Qiuchen Wang, Hongmin Zhan, Jinxin Bian, Xingchao Li, Peijin Zheng, Hao Cheng, Pengjun Xie, Kaipeng Zhang, Jiawei Liu, and Zheng-Jun Zha. ARISE-RL: Agentic rubric-grounded iterative self-evolution with reinforcement learning, 2026a. URL https://arxiv.org/abs/2609.01058v1.

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. In *The Fourteenth International Conference on Learning Representations*, 2026b. URL https://openreview.net/forum?id=pBjy4ek2QV.

Yuxiang Zheng, Dayuan Fu, Xiangkun Hu, et al. DeepResearcher: Scaling deep research via reinforcement learning in real-world environments. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 414–431, Suzhou, China, November 2025. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.22. URL https://aclanthology.org/2025.emnlp-main.22/.

Junyi Zhou, Qiyuan Zhang, Yufei Wang, Fuyuan Lyu, Yidong Ming, Can Xu, Qingfeng Sun, Kai Zheng, Peng Kang, Xue Liu, and Chen Ma. RubricBench: Aligning model-generated rubrics with human standards. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 31179–31200, San Diego, California, United States, July 2026a. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1439. URL https://aclanthology.org/2026.acl-long.1439/.

Junyi Zhou, Qiyuan Zhang, Yufei Wang, Fuyuan Lyu, Yidong Ming, Can Xu, Qingfeng Sun, Kai Zheng, Peng Kang, Xue Liu, and Chen Ma. RubricBench: Aligning model-generated rubrics with human standards. In Maria Liakata, Viviane P. Moreira, Jiajun Zhang, and David Jurgens (eds.), *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 31179–31200, San Diego, California, United States, July 2026b. Association for Computational Linguistics. ISBN 979-8-89176-390-6. doi: 10.18653/v1/2026.acl-long.1439. URL https://aclanthology.org/2026.acl-long.1439/.

Minghang Zhu, Chuyang Wei, Junhao Xu, Yilin Cheng, Zhumin Chen, and Jiyan He. DeepRubric: Evidence-tree rubric supervision for efficient reinforcement learning of deep research agents, 2026. URL https://arxiv.org/abs/2606.17029.

CONTENTS

A Use of Large Language Models (LLMs) **17**

B Related Work **17**

C Problem Formulation for Scope-First Rubric Construction **18**

C.1 Task Requirements and Evaluation Scope 18

C.2 Scope Misspecification 18

C.3 Scope-Preserving Rubric Construction 18

C.4 Hierarchical Scoring and Reward Aggregation 18

D Theoretical Analysis **19**

D.1 Scope as a Task-Oriented Evaluation Representation 19

D.2 Stage-I Quality: Coverage, Necessity, and Importance 20

D.3 Stage-II Refinement at a Fixed Evaluation Budget 20

D.4 Why Satisfaction-Gated Refinement Must Keep the Scope Fixed 21

D.5 Why Item-Level Validity Is Insufficient 22

D.6 Scope Error and Scoring Error 22

D.7 Preservation of Direction Weights and Duplicate-Counting Control 23

E Qualitative Analysis and Case Studies **23**

E.1 Case Selection and Reading Guide 24

E.2 Evaluation Direction Quality 24

E.3 Within-Direction Rubric Set Quality 24

E.4 Cross-Direction Rubric Set Quality 25

E.5 Agent Response Comparison and Failure Cases 25

F Data Creation Details **25**

G Training Details **26**

H Experimental Details **26**

I More Results and Analysis **36**

I.1 Complete Human Preference Alignment Results 36

I.2 Complete Downstream Agent Results 36

I.3 Individual Rubric Quality 38

I.4 Additional Direction and Rubric-Set Analyses 38

I.5 Additional Ablation Results 39

I.6 Full Training Curves and Cost Analysis 39

I.7 Cross-Domain Generalization 39

J Analysis on Searched Tools and Domain Distributions **39**

K Discussion and Future Work **40**

A USE OF LARGE LANGUAGE MODELS (LLMS)

In addition to their roles in the experiments, large language models assisted with manuscript preparation and literature organization. The assistance covered the following activities:

1. **Writing and editing.** Organizing the appendix, drafting and revising English prose, formalizing definitions and drafting mathematical arguments, and checking consistency between the manuscript structure and its cross-references.
2. **Literature search and organization.** Locating relevant papers and organizing comparisons of their problem formulations, analyses, and experimental reporting.
3. **Manuscript implementation.** Preparing LaTeX source and checking document compilation and layout.

The authors are responsible for the final text, mathematical arguments, citations, and empirical claims. Experimental model roles and configurations are described in Appendices G and H.

B RELATED WORK

B.1 LONG-FORM DEEP RESEARCH AGENTS AND EVALUATION

Deep research agents combine information retrieval with evidence synthesis and report generation. Systems such as WebThinker, WebWeaver, and Tongyi DeepResearch study how to organize these operations over extended interaction trajectories (Li et al., 2025; 2026d; Tongyi DeepResearch Team et al., 2025). Evaluating the resulting reports requires judgments about what the task asks for, whether the evidence supports the claims, and how the report integrates its findings. DeepResearch Bench, ResearchRubrics, and ResearchQA provide complementary evaluation settings for general research reports and scientific synthesis (Du et al., 2026; Sharma et al., 2026; Yifei et al., 2026). These settings motivate query-specific evaluation: the importance of a particular fact, comparison, or limitation depends on the research question.

S2R addresses the construction of the criteria used to evaluate a report. Its evaluation scope specifies task-level directions, their inclusion boundaries, and their relative priorities before individual rubrics are generated. This intermediate representation connects the query to the final scoring criteria and provides a separate object for diagnosing missing or unsupported requirements.

B.2 RUBRIC-BASED REWARD MODELING

Reward models translate response quality into a signal for selection or optimization. Scalar reward models provide a numerical assessment, while reasoning-based reward models expose intermediate reasoning before their judgment (Wang et al., 2024; Chen et al., 2026a). Rubric-based rewards further decompose evaluation into criteria that can be checked against individual responses (Gunjal et al., 2026). This decomposition makes the connection between a response feature and its contribution to the reward inspectable.

For deep research, the construction process can use task context, retrieved evidence, candidate reports, and feedback from policy training. DR Tulu combines deep-research training with evolving rubrics, while query-specific rubric generators learn criteria tailored to individual requests (Shao et al., 2026; Lv et al., 2026). S2R separates the recovery of the evaluation scope from the generation of checkable items. Its generator is training-free; the generated rubrics can subsequently serve as rewards for a trainable agent. Generator training, rubric updating, and policy optimization are therefore separate comparison axes.

B.3 RUBRIC CONSTRUCTION AND QUALITY CONTROL

Rubric construction methods differ in the information they use and the level at which they revise an evaluation. Retrieval can supply relevant criteria, while decomposition and iterative refinement can improve the specificity of an initial set (Dhole & Agichtein, 2026; Liu et al., 2024). An individual criterion can be well grounded and easy to judge while the complete set still omits a necessary requirement or evaluates one behavior repeatedly. This distinction motivates separate checks of item validity, within-direction coverage, and cross-direction overlap.

In S2R, the scope fixes the content and weight of each direction, and rubric construction operationalizes that direction within its boundary. The resulting hierarchy makes two questions explicit: whether the recovered directions represent the task, and whether the rubrics faithfully implement those directions. Appendix D analyzes these two sources of evaluation error, while Appendix E organizes the corresponding qualitative diagnostics.

C Problem Formulation for Scope-First Rubric Construction

C.1 Task Requirements and Evaluation Scope

Let q denote an LFDR query and y a candidate research report. Task requirements specify the content, evidence, and reasoning needed to answer q. An evaluation direction $d_i = (c_i, b_i)$ groups related judgments into one task-level facet: c_i specifies its content and b_i specifies its inclusion boundary. As in Section 3.1, the query-specific evaluation scope is

$$\mathcal{S}(q) = \{(d_i, w_i)\}_{i=1}^{m}, \qquad w_i \geq 0, \quad \sum_{i=1}^{m} w_i = 1. \tag{8}$$

The weights express relative task priorities. A process stage such as retrieval or evidence analysis supplies a cue for discovering a requirement; it becomes an evaluation direction only when the query makes that requirement relevant to the final report.

For each direction, $R_i = \{r_{ij}\}_{j=1}^{n_i}$ is a set of checkable rubric items. Each item operationalizes part of c_i within b_i. The construction follows the dependency

$$q \longrightarrow \mathcal{S}(q) \longrightarrow R(q) = \{R_i\}_{i=1}^{m} \longrightarrow S(y \mid q). \tag{9}$$

The first mapping determines what matters for the task; the second makes those requirements checkable; the last evaluates a response. This ordering allows a change in rubric granularity without automatically changing the priority assigned to its direction.

C.2 Scope Misspecification

We distinguish three ways in which an evaluation can depart from the task.

Task-Coverage Incompleteness. A necessary requirement falls outside the content and boundaries of all recovered directions. For example, a query may require comparison across settings, while the recovered scope includes only separate factual descriptions of each setting. If the comparison already belongs to a direction but lacks a corresponding rubric, the failure is instead within-direction rubric-set incompleteness. The former requires scope repair; the latter requires faithful operationalization of an existing direction.

Task-Requirement Overreach. An optional or unsupported detail becomes a condition for a high score. For example, a reference report may adopt one particular organization of the evidence. Requiring every response to follow that organization would promote a reference-specific choice into a task requirement.

Task-Priority Misalignment. The effective importance of a requirement differs from its task priority. If a direction receives more rubric items solely because it admits finer decomposition, a flat average can give that direction more influence even though the query has not changed.

These examples illustrate distinct evaluation failures. Coverage concerns missing information, overreach concerns unsupported inclusion, and priority misalignment concerns the distribution of scoring weight. Their empirical diagnostics are specified in Appendix H.2.

C.3 Scope-Preserving Rubric Construction

Given q and a set of reference reports $\mathcal{H}$, S2R first recovers explicit directions, then task-dependent implicit directions, and finally supplemental directions supported by the references. A supplemental candidate is retained when it is necessary for the task, supported across reference reports, and absent from the existing scope. A fact appearing in a reference is thus a candidate source of evidence for an omission, with task necessity determining whether it belongs in the evaluation.

Task-coverage, necessity, and priority analyses review the candidate set. Once the scope is fixed, generation and quality control preserve $\{c_i, b_i, w_i\}_{i=1}^{m}$. An item-level edit can sharpen a pass condition; a within-direction edit can repair a missing judgment or merge repetitions; a cross-direction edit can correct item ownership. A change that requires a new direction or a different direction weight is a scope-level revision and must be distinguished from these local edits.

C.4 Hierarchical Scoring and Reward Aggregation

We use the following aggregation model to analyze scope preservation. For direction i, let $\mathcal{G}_i$ be a finite set of scoring units. A unit represents either an individual item or a group of items assigned one aggregate score and weight. Let $v_{ig} \geq 0$ denote its base weight, with $\sum_{g \in \mathcal{G}_i} v_{ig} = 1$. For

report y, let $\mathcal{A}_i(y) \subseteq \mathcal{G}_i$ contain the applicable units and $z_{ig}(y) \in [0,1]$ their scores. Assuming $V_i(y) = \sum_{g \in \mathcal{A}_i(y)} v_{ig} > 0$, define

$$s_i(y) = \frac{\sum_{g \in \mathcal{A}_i(y)} v_{ig} z_{ig}(y)}{V_i(y)}, \qquad S(y \mid q) = \sum_{i=1}^{m} w_i s_i(y). \tag{10}$$

Here s_i measures satisfaction within a direction, and w_i fixes that direction's contribution to the total score. The positivity condition holds, for example, if every direction contains a positive-weight unit that remains applicable to every response. A prerequisite-based rule may exclude a dependent unit when its prerequisite fails; the remaining weights are then renormalized within the same direction.

D Theoretical Analysis

We analyze the evaluation function for a fixed query. The results separate what can be guaranteed by the representation and construction rules from what must be measured on data. In particular, the comparisons below are conditional on explicit assumptions about requirement recovery, judge noise, and evaluation cost; they do not turn a construction assumption into an empirical result.

D.1 Scope as a Task-Oriented Evaluation Representation

Let $\mathcal{U}_q = \{u_1, \ldots, u_K\}$ be the atomic judgment points that define the target evaluation for query q. Each u_k has a task-valid satisfaction function $t_k(y) \in [0,1]$ and a target importance $\alpha_k \geq 0$, with $\sum_k \alpha_k = 1$. The target task utility is

$$S_q^\star(y) = \sum_{k=1}^{K} \alpha_k t_k(y). \tag{11}$$

An evaluation scope is a task-conditioned representation when it provides a partition map $\kappa : \mathcal{U}_q \to \{1, \ldots, m\}$, together with the semantics, inclusion boundary, requirement relations, and direction weight for every direction. A scope is *valid* if each requirement is assigned to exactly one direction, its polarity and boundary are preserved, and

$$w_i = \sum_{k : \kappa(u_k) = i} \alpha_k. \tag{12}$$

The map is a representation of the task requirements rather than a list of response features: changing the response cannot change κ, the boundaries, or the direction weights.

Proposition D.1 (Scope is a sufficient task-oriented evaluation structure) *Suppose a valid scope $\mathcal{S}_q$ is given and every rubric generated under direction i is a boundary-preserving decomposition of the atomic points $\{u_k : \kappa(u_k) = i\}$ for every direction with $w_i > 0$. If the within-direction rubric weights sum to one, then the resulting hierarchical score can be written as $S_q(y) = F(\mathcal{S}_q, y)$ and equals the target utility in Eq. equation 11. Consequently, all information needed for task-conditioned evaluation is available through the factorization*

$$q \longrightarrow \mathcal{U}_q \longrightarrow \mathcal{S}_q \longrightarrow \mathcal{R}_q \longrightarrow S_q(y). \tag{13}$$

Proof. For direction i, let each rubric r decompose into atomic points with nonnegative coefficients β_{rk} such that $\sum_{r \in \mathcal{R}_{q,i}} \alpha_r \beta_{rk} = \alpha_k / w_i$ for every k assigned to i. Boundary preservation means that no coefficient is assigned to a point outside that direction. Substituting this decomposition into the hierarchical score and exchanging the finite sums gives

$$\sum_i w_i \sum_{r \in \mathcal{R}_{q,i}} \alpha_r \sum_{k : \kappa(u_k) = i} \beta_{rk} t_k(y) = \sum_k \alpha_k t_k(y) = S_q^\star(y).$$

Thus the scope stores the task-dependent content, boundaries, relations, and priorities that the rubric generator must preserve; the rubric set only chooses a checkable granularity inside that structure. ■

The proposition gives a precise meaning to "scope-first." It does not claim that a recovered scope is automatically valid. Validity is testable through requirement coverage, direction correctness, necessity, and weight alignment; those diagnostics are defined in Appendix H.2.

D.2 Stage-I Quality: Coverage, Necessity, and Importance

We compare two construction procedures on the same query and evidence: scope-first ($M = S$) and direct rubric-first ($M = R$). Let $X_{M,k} = 1$ when the final rubric set contains a valid item for u_k; validity includes the required semantics, boundary, and polarity. Let V_M be the total weight assigned to unsupported or task-irrelevant points, and let $D_M = \frac{1}{2}\|\widehat{\alpha}_M - \alpha\|_1$ be the total-variation error of the recovered importance vector after aligning points on $\mathcal{U}_q \cup \widehat{\mathcal{U}}_M$. We use the following measurable quality quantities:

$$\mathrm{Cov}(M) = \sum_k \alpha_k X_{M,k}, \qquad \mathrm{Nec}(M) = 1 - V_M, \qquad \mathrm{Imp}(M) = 1 - D_M. \tag{14}$$

Coverage counts only task-valid requirement points, so it is correctness- weighted: a wrong semantic, boundary, or polarity match receives no credit. Necessity penalizes unsupported additions, and importance measures whether the scoring mass is placed on the right requirements. A weighted composite is $Q_1(M) = \lambda_C \mathrm{Cov}(M) + \lambda_N \mathrm{Nec}(M) + \lambda_I \mathrm{Imp}(M)$ for nonnegative coefficients summing to one.

Proposition D.2 (Scope-first improves Stage-I quality under explicit correction) *Assume the direct rubric-first procedure omits u_k with probability o_k, assigns unsupported mass v, and has importance error D_R. Suppose the scope recovery and review stage catches an omitted requirement u_k with probability $\rho_k \in [0,1]$, removes each unsupported mass element with probability at least $\tau \in [0,1]$, and reduces importance error to at most $(1-\eta)D_R$ for some $\eta \in [0,1]$. Assume these corrections do not introduce new omissions or unsupported mass. Then*

$$\mathbb{E}[\mathrm{Cov}(S) - \mathrm{Cov}(R)] = \sum_k \alpha_k \rho_k o_k \geq 0, \tag{15}$$

$$\mathbb{E}[\mathrm{Nec}(S) - \mathrm{Nec}(R)] \geq \tau\, \mathbb{E}[V_R] \geq 0, \tag{16}$$

$$\mathbb{E}[\mathrm{Imp}(S) - \mathrm{Imp}(R)] \geq \eta\, \mathbb{E}[D_R] \geq 0. \tag{17}$$

Therefore $\mathbb{E}[Q_1(S)] \geq \mathbb{E}[Q_1(R)]$, with a strict inequality whenever a positively weighted correction is active.

Proof. The scope review leaves omission probability $(1-\rho_k)o_k$, so the expected coverage difference is the first expression in Eq. equation 17. Removing at least a τ fraction of unsupported mass increases necessity by at least τV_R. The assumed weight correction gives $D_S \leq (1-\eta)D_R$, hence $\mathrm{Imp}(S) - \mathrm{Imp}(R) = D_R - D_S \geq \eta D_R$. Taking the same nonnegative linear combination proves the result. ∎

The proposition identifies what the Stage-I evidence must establish rather than assuming that scope-first is universally superior. On a held-out set of queries, ρ_k is estimated from strict requirement coverage, τ from necessity and overreach labels, and η from the aligned importance error. Direction correctness and cross-direction redundancy can be reported as additional constraints on the same comparison. If the review does not improve these quantities, the proposition yields no strict advantage.

D.3 Stage-II Refinement at a Fixed Evaluation Budget

Let X_A denote the information returned by a complete but static rubric set and let X_F denote the information after valid response-aware refinement within the fixed scope. A refinement is valid when it only decomposes an existing judgment point and keeps its direction, boundary, polarity, and weight budget. We write $\sigma(X)$ for the information generated by an observation X; score reuse makes $\sigma(X_A) \subseteq \sigma(X_F)$. For the target utility $T = S_q^\star(Y_t)$ on the current rollout distribution, the best squared-loss predictors based on these observations are conditional expectations. Define the refinement gain

$$G_t = \mathbb{E}\Big[\big(\mathbb{E}[T \mid X_F] - \mathbb{E}[T \mid X_A]\big)^2\Big] \geq 0. \tag{18}$$

The gain is strictly positive exactly when the refined children change the conditional mean of T relative to X_A on a set of positive probability; this is the measurable form of an unresolved task-relevant difference. In the method notation, LowDisc locates a candidate, while $\Delta_r^t \neq \emptyset$ is the additional condition that can make $G_t > 0$.

Let $\widehat{T}_A$ and $\widehat{T}_F$ add zero-mean independent evaluator noise with variances σ_A^2 and σ_F^2, respectively. Let C_A and C_F include generation, judge calls, and tokens. If $c(\mathcal{C}_r)$ is the cost of evaluating a child

set and C_{saved} is the cost of reusing unchanged scores, then the per-rollout change is

$$C_F - C_A = \sum_{r \in \mathcal{A}_t(q)} \big(c(\mathcal{C}_r^t) - c(r)\big) - C_{\text{saved}}. \tag{19}$$

We compare methods under the same budget B: $C_A \le B$ and $C_F \le B$ (equality is allowed by reallocating saved calls to the active candidates).

Proposition D.3 (Selective refinement improves reward fidelity at equal cost) *Under the budget condition above, if*

$$G_t > \sigma_F^2 - \sigma_A^2, \tag{20}$$

then

$$\mathbb{E}[(T - \widehat{T}_F)^2] < \mathbb{E}[(T - \widehat{T}_A)^2]. \tag{21}$$

If $G_t = 0$ and refinement does not reduce evaluator noise ($\sigma_F^2 \ge \sigma_A^2$), refinement cannot improve squared-loss fidelity and incurs only its additional evaluation cost. Therefore the trigger used in Eq. equation 4 is necessary for a strict cost-normalized improvement: low discriminability alone is insufficient; there must also be an unresolved task-relevant difference.

Proof. The law of total variance and the nesting $\sigma(X_A) \subseteq \sigma(X_F)$ give

$$\mathbb{E}[(T - \mathbb{E}[T \mid X_A])^2] - \mathbb{E}[(T - \mathbb{E}[T \mid X_F])^2] = G_t.$$

Adding independent evaluator noise yields $\text{MSE}_A - \text{MSE}_F = G_t - (\sigma_F^2 - \sigma_A^2)$, which is positive under Eq. equation 20. When $G_t = 0$, the conditional predictors are equally informative, so extra calls cannot increase reward fidelity. ■

This result compares information per evaluation budget, not the number of rubric strings. A static all-atomic baseline may spend calls on criteria that are already uniformly satisfied or already separate the current responses. The fixed-scope method reuses those scores and spends the same budget on children whose conditional variance is nonzero. The quantities in Eq. equation 20 are measurable through held-out response pairs, rubric-level score variance, judge repeatability, and actual API token cost.

For reporting, the cost-normalized reward quality can be written as $\text{Eff}(M) = (1 - \text{MSE}_M)/C_M$, and total training expense as $C_{\text{total}} = C_{\text{Stage I}} + \sum_t C_t$. Stage-I scope recovery may have a larger one-time construction cost; a claim of lower total cost therefore requires reporting that term separately. Equation equation 19 gives the testable condition for Stage-II savings from selective refinement and score reuse.

D.4 Why Satisfaction-Gated Refinement Must Keep the Scope Fixed

A response-driven evolving rubric can change which task requirements receive weight. Let $\widehat{\alpha}_t$ be its effective importance vector and define its scope drift by

$$\text{Drift}_t = \frac{1}{2}\|\widehat{\alpha}_t - \alpha\|_1. \tag{22}$$

Scope-constrained refinement keeps direction weights and boundaries fixed, so its direction-level drift is zero; only within-direction granularity can change. The error decomposition in Proposition D.6 implies the certified pointwise bound

$$|\widehat{S}_t(y) - S_q^\star(y)| \le \text{Drift}_t + \sum_k \widehat{\alpha}_{t,k} |\widehat{t}_{t,k}(y) - t_k(y)|. \tag{23}$$

Proposition D.4 (Response-only evolution has an unavoidable drift failure mode) *If a response-driven update produces $\text{Drift}_t = \delta > 0$ and the task admits a report whose satisfaction vector separates the positive and negative drift coordinates, then there exists a satisfaction vector $t \in [0,1]^K$ for which its structural score error is at least δ, even with a perfect judge. A fixed-scope refinement has zero structural error under the same realizability assumption. Hence response-only evolution cannot provide a uniformly better guarantee unless it enforces $\text{Drift}_t = 0$ or receives an independent task-necessity correction.*

Proof. Let $\beta = \widehat{\alpha}_t - \alpha$. Because both vectors sum to one, the positive and negative masses of β each equal $\|\beta\|_1/2 = \delta$. Choose a realizable response with $t_k = 1$ on the positive coordinates of β and $t_k = 0$ on the negative coordinates. Then $|\widehat{\alpha}_t^\top t - \alpha^\top t| = \beta^\top t = \delta$. The fixed-scope construction preserves α at the direction level, so the same structural term is zero; only judge and within-scope decomposition errors remain. ∎

Thus satisfaction is used as a trigger for granularity, not as permission to rewrite the task. The measurable comparison should report reward MSE, pairwise preference accuracy, paired score separation d, scope drift, requirement coverage, and total judge tokens. The same distinction carries to policy optimization. Let $J(\pi) = \mathbb{E}_{q,y\sim\pi}[S_q^\star(y)]$ and let $\widehat{\pi}$ be a policy whose estimated-reward suboptimality is at most ξ. If $\sup_{q,y} |\widehat{S}(q,y) - S_q^\star(y)| \le \varepsilon$, then

$$J(\pi^\star) - J(\widehat{\pi}) \le 2\varepsilon + \xi, \tag{24}$$

where $\pi^\star$ maximizes J. Indeed, add and subtract the estimated reward of $\pi^\star$ and $\widehat{\pi}$ and apply the uniform error bound twice. Consequently, any refinement that lowers the reward error from ε_A to ε_F at the same cost tightens the worst-case task-utility gap by $2(\varepsilon_A - \varepsilon_F)$, while scope drift directly enlarges the bound through Eq. equation 23. This is the formal sense in which a higher-quality rubric-as-reward can yield an agent with higher task-grounded ability: it supplies a smaller optimization target error, not merely a more detailed rubric description.

D.5 Why Item-Level Validity Is Insufficient

Let $\phi_R(y)$ be the vector of scores returned by a rubric set R and $S_R(y) = F(\phi_R(y))$ any deterministic aggregation of that vector. Let $S^\star(y)$ be the target evaluation for the query.

Proposition D.5 (Indistinguishability under omitted requirements) *If two reports y_a, y_b satisfy $\phi_R(y_a) = \phi_R(y_b)$ but $S^\star(y_a) \neq S^\star(y_b)$, then $S_R(y_a) = S_R(y_b)$. Writing $\delta = |S^\star(y_a) - S^\star(y_b)|$, at least one report has absolute evaluation error at least $\delta/2$.*

Proof. Identical score vectors are identical inputs to F, so both reports receive the same score c. By the triangle inequality,

$$\delta \le |S^\star(y_a) - c| + |c - S^\star(y_b)|.$$

At least one term is therefore at least $\delta/2$. ∎

Interpretation. An evaluation cannot recover a distinction that its input scores do not represent. For a minimal example, suppose the target score equally weights two requirements and R measures only the first. Reports with satisfaction vectors $(1,1)$ and $(1,0)$ tie under R although their target scores are 1 and $1/2$. Reweighting the existing score cannot recover the omitted distinction. The proposition applies both to a requirement omitted from the scope and to an existing direction that is incompletely operationalized. These cases motivate recovering the necessary directions and then checking whether their rubric sets retain the required distinctions.

D.6 Scope Error and Scoring Error

For the analysis, assume that the target and constructed evaluations admit an additive representation over a common finite set of judgment points $\mathcal{U} = \{u_1, \ldots, u_K\}$. Each point includes the semantic judgment and its inclusion and applicability conditions. Judgments with different boundaries are distinct points even if they concern the same topic. This representation aligns judgments by these definitions, rather than by direction names. It can be formed by matching equivalent judgment points, retaining unmatched points from either evaluation, and assigning zero weight where a point is absent. If a criterion combines several points, the additive representation requires an explicit decomposition of its score and weight.

Let $\alpha, \widehat{\alpha} \in \mathbb{R}^K$ be nonnegative vectors summing to one, and let $t(y), \widehat{t}(y) \in [0,1]^K$ be the target and estimated satisfaction vectors on this same index set. Define

$$S^\star(y) = \alpha^\top t(y), \qquad \widehat{S}(y) = \widehat{\alpha}^\top \widehat{t}(y). \tag{25}$$

An omitted requirement has $\alpha_k > 0$ and $\widehat{\alpha}_k = 0$; an unsupported requirement has $\alpha_k = 0$ and $\widehat{\alpha}_k > 0$. Different allocations to retained points encode priority mismatch.

Proposition D.6 (Scope and scoring error bound) *Under the representation above, for every report y,*

$$|\widehat{S}(y) - S^\star(y)| \le \frac{1}{2}\|\widehat{\alpha} - \alpha\|_1 + \sum_{k=1}^{K} \widehat{\alpha}_k |\widehat{t}_k(y) - t_k(y)|. \tag{26}$$

Proof. Adding and subtracting $\widehat{\alpha}^\top t(y)$ gives

$$\widehat{S}(y) - S^\star(y) = (\widehat{\alpha} - \alpha)^\top t(y) + \widehat{\alpha}^\top (\widehat{t}(y) - t(y)).$$

Put $\beta = \widehat{\alpha} - \alpha$. Since $\sum_k \beta_k = 0$, its positive and negative masses both equal $\|\beta\|_1/2$. The bounds $0 \le t_k(y) \le 1$ imply

$$-\frac{1}{2}\|\beta\|_1 \le \sum_k \beta_k t_k(y) \le \frac{1}{2}\|\beta\|_1.$$

Applying the triangle inequality to the decomposition and using $\widehat{\alpha}_k \ge 0$ proves Equation 26. ■

Ranking consequence. Suppose the right-hand side of Equation 26 is bounded by ϵ_a and ϵ_b for y_a and y_b, respectively. If $S^\star(y_a) - S^\star(y_b) > \epsilon_a + \epsilon_b$, then $\widehat{S}(y_a) > \widehat{S}(y_b)$. Indeed,

$$\widehat{S}(y_a) - \widehat{S}(y_b) \ge S^\star(y_a) - S^\star(y_b) - \epsilon_a - \epsilon_b > 0.$$

The familiar 2ϵ margin follows when both errors are at most ϵ.

Interpretation. The first term in the bound measures a mismatch in the allocation of evaluation weight, while the second measures satisfaction-estimation error under the constructed allocation. Scope recovery targets the former; rubric construction and judging also affect the latter. The bound motivates examining both objects. Empirical coverage and necessity metrics are diagnostics of scope quality, not direct observations of the latent α or t.

For response-dependent applicability, the bound holds pointwise whenever the scores admit Equation 25, with $\alpha(y)$ and $\widehat{\alpha}(y)$ if the effective allocations depend on y. For each report, both vectors must be nonnegative, sum to one, and use the same index set $\mathcal{U}$. Fixed direction weights alone do not imply fixed weights for all atomic judgment points after applicability-based normalization.

D.7 Preservation of Direction Weights and Duplicate-Counting Control

Consider Equation 10 with nonnegative within-direction weights and $V_i(y) > 0$.

Proposition D.7 (Fixed direction budgets) *For every report y, the effective coefficients of all applicable scoring units in direction i sum to w_i. Consequently, $0 \le S(y \mid q) \le 1$.*

Proof. The coefficient of unit $g \in \mathcal{A}_i(y)$ is $a_{ig}(y) = w_i v_{ig}/V_i(y)$. Hence

$$\sum_{g \in \mathcal{A}_i(y)} a_{ig}(y) = w_i.$$

All coefficients are nonnegative, and their sum over directions equals $\sum_i w_i = 1$. The total score is therefore a convex combination of unit scores in $[0, 1]$. ■

Duplicate representations. Suppose adding a duplicate rubric leaves the scoring-unit set, unit assignments, applicability, base weights, and unit scores unchanged. Both the numerator and denominator in Equation 10 are then unchanged, so the overall score is invariant. A count-once representation with a stable judgment identity satisfies this condition when the duplicate is mapped to its existing unit without altering its score.

The condition concerns effective scoring units. Averaging raw item scores within a cluster generally lacks the same invariance: duplicating the score 1 in a cluster with scores $(0, 1)$ changes its mean from $1/2$ to $2/3$. Likewise, normalization preserves a direction's total budget but does not by itself ensure coverage or correct the relative weights of distinct judgments within that direction. These properties motivate separate within- and cross-direction quality checks.

E Qualitative Analysis and Case Studies

The qualitative analysis follows the evaluation chain from a task requirement to its direction, rubric items, response evidence, and score. Direction quality concerns which requirements enter the scope; within-direction quality concerns how one direction is operationalized; cross-direction quality concerns ownership and repeated evaluation.

Concrete ResearchQA case. We instantiate the comparison with the ResearchQA preference pair `cv_plurality_preferences:17442129294819-s4:gpt:vs:gemini`, whose query is "How do different levels of CNN features affect the performance of deep facial attribute analysis methods?" Human preference is A. The archived query-level comparator is a rubric-first full-rubric control used to expose the construction contrast. The available trace contains its construction and judge records, whereas Table 2 contains downstream agent results.

On this pair, Scope2Rubric recovers four evaluation directions and seven rubrics, and selects A (normalized scores 100.0 vs. 93.74). The full-rubric control produces ten flat direction groups and 39 rubrics, and selects B (81.40 vs. 84.96). Thus the same query and responses yield opposite decisions: Scope2Rubric agrees with the human preference, while the rubric-first control does not.

E.1 Case Selection and Reading Guide

The two main-text panels share a query so that rubric changes and agent behavior can be examined against the same task requirements. The full record consists of the query, reference reports used in construction, candidate and retained directions, initial and reviewed rubric sets, direction and item weights, candidate responses, and criterion-level scores. Stable identifiers connect each requirement to its direction, rubrics, and supporting response spans.

The rubric comparison holds the two candidate responses fixed. The case is selected because the disagreement is inspectable: A stays on facial attribute analysis, whereas B opens with a valence/arousal emotion-recognition result and a model-specific string. This makes it possible to trace how a task-level scope changes the evidence that receives weight, without attributing the outcome to a different query or response pair.

E.2 Evaluation Direction Quality

The direction analysis begins with the requirements expressed in the query and the implicit requirements needed to answer it. For this case the four retained directions are: (D1) directly relate feature level to facial- attribute-analysis performance; (D2) distinguish shallow, middle, and deep feature semantics and align them with attribute types; (D3) explain the continuous path from level differences through task matching to performance; and (D4) provide minimal evidence and condition boundaries without claiming that one level is universally optimal. Each direction has an explicit coverage boundary, so the emotion-recognition detour in B is outside D1–D4. The retained weights are $0.2167, 0.2167, 0.2167, 0.3500$, respectively.

Reference-supported additions require a task-level justification. A repeated detail across reports can reveal a missing requirement, but its frequency alone does not establish necessity or priority. The final comparison therefore examines both recovered requirements and excluded out-of-scope details. Direction weights are interpreted through the role of the corresponding requirement and the consequence of omitting it. The requirement–direction heatmap retains the identity, condition, and polarity of each expert requirement. A cell shows whether the direction explicitly includes that evaluation target; a broad heading alone is not sufficient. Correct, incorrect, and insufficient-evidence directions are illustrated with their supporting or missing evidence. This connects the aggregate scope-coverage score to an inspectable case. The flat control instead exposes ten direction groups (E1–E3, D1–D6, and M1) before scoring. Its groups separately enumerate topic boundaries, hierarchical semantics, task matching, fusion, method-setting caveats, analysis structure, and a scale-matching framework. These checks can all be reasonable in isolation, but the trace does not first decide which four query-specific directions own the judgments. The case therefore contrasts scope recovery with a larger checklist, rather than checklist length with checklist length.

E.3 Within-Direction Rubric Set Quality

For a fixed direction, the comparison begins with its content and boundary, then aligns the rubric set to the necessary judgment points. The final sets contain one rubric under D1, two under D2, two under D3, and two under D4. Representative checks ask whether an answer (i) directly relates feature levels to performance, (ii) distinguishes level semantics and attribute dependence, (iii) gives a level–task–performance explanation chain and a conditional synthesis, and (iv) supplies evidence and applicability bounds. This alignment distinguishes coverage repair from rewriting: a missing judgment expands the scope, while a split or merge within a retained direction changes granularity without changing what is being evaluated.

Each edit is paired with the response evidence it evaluates. Pass/fail conditions distinguish substantive satisfaction from a mention of the relevant topic. Criterion scores and the resulting direction score make the effect of the edit visible. The direction's fixed weight provides the reference point

for assessing whether finer decomposition has changed its influence on the total evaluation. The seven rubrics are normalized within their directions and the four direction weights are kept fixed. In contrast, the control's 39 items are scored as a flat full set. The larger set contains several overlapping ways to reward general layer descriptions, fusion suggestions, and scale terminology; it does not provide a separate ownership boundary for the query's central task. The resulting score reversal is therefore visible at the set level even though no individual score is treated as a causal ablation.

E.4 Cross-Direction Rubric Set Quality

The cross-direction comparison places adjacent direction boundaries and their rubric items side by side. It distinguishes three relations: two items testing the same judgment, different judgments sharing a prerequisite, and an item assigned outside its intended direction. These relations call for different edits. Repetition can be merged or narrowed; misplaced content can be reassigned; a legitimate prerequisite relation requires an explicit applicability or aggregation rule.

The analysis follows the shared response evidence through the original and reviewed assignments and compares its effective scoring weight. Semantic similarity diagnoses overlapping content. Scoring dependence requires score-level evidence, such as the controlled probes described in Appendix H.2. An embedding visualization can help locate related items, while the judgment and score records explain whether they produce repeated credit or penalty. The most diagnostic cross-direction relation is between D2 and D3. D2 owns what shallow, middle, and deep features represent and which attribute types need them; D3 owns how that alignment changes performance. Keeping these boundaries separate prevents a long list of feature facts from receiving credit twice. The control's E/D/M groups distribute the same content over multiple flat checks, including separate generic fusion and scale-matching items, so repeated evidence can contribute without first being assigned to a single query-specific direction.

E.5 Agent Response Comparison and Failure Cases

The response comparison connects a concrete passage to the criterion that evaluates it. Relevant differences include whether a claim has supporting evidence, whether a cited source entails the associated statement, and whether the report synthesizes the sources needed for the query. The agent checkpoint and generation settings accompany each response.

A failure case is localized to the same chain. A states that lower layers capture edges and local information useful for locally defined attributes, whereas deeper layers capture semantic information for complex attributes; it then returns to facial attribute analysis and multi-label learning. B's opening instead says that low- and high-level features affect "valence and arousal" and names `CNN-3RNN-2nd-pool_last-pool_fc`. The latter is an emotion-recognition result rather than evidence about the queried facial attribute-analysis task. Scope2Rubric's D1 and D4 boundaries therefore retain the task-aligned evidence while making the detour non-creditable. The control's score (84.96 for B versus 81.40 for A) reverses the human ordering, which is the concrete failure shown in Figure 9.

F Data Creation Details

F.1 Query Sources and Selection

S2R takes a research query as the input to rubric construction. The experiments use separate data roles for human preference alignment, downstream agent evaluation, direction and rubric diagnostics, and expert-rubric alignment. Appendix H.1 maps the named benchmarks to these roles. A query can be associated with multiple candidate reports or preference pairs; these records remain linked to their originating query.

F.2 Reference Report Collection

Let $\mathcal{H}(q) = \{h_1, \ldots, h_L\}$ denote the reference reports available for a query. The recovery stage examines these reports individually for requirements absent from the explicit and implicit directions, then groups related candidates across reports. The references provide evidence for diagnosing omissions; acceptance also requires task necessity and noncoverage by the current direction set.

F.3 Rubric Construction Records

The construction has three successive representations: candidate directions, a fixed weighted scope, and direction-specific rubric sets. Each direction carries content, a boundary, and a weight; each item

carries a checkable judgment, pass/fail conditions, and an initial within-direction weight. Quality-control records connect an original item to the corresponding retained, merged, split, removed, or reassigned item.

F.4 Preference and Training Data Preparation

Preference evaluation associates a query with candidate reports and an ordering supplied by its evaluation data. Policy training instead uses queries to generate agent rollouts and obtains a reward from rubric-based judgments. Expert rubrics supply a third role: a reference evaluation against which generated criteria or response rankings can be compared. These roles determine which records are inputs, supervision, or evaluation references in each experiment.

G Training Details

G.1 Starting Checkpoint and Policy Setup

The downstream policy starts from Qwen3-8B-SFT. S2R constructs the rubrics without training its generator, and the downstream experiment optimizes the agent using rubric-based rewards. The GPT5 and Gemini3 result labels distinguish rubric-generation variants.

G.2 Reward Computation

Gemini-2.5-Pro serves as the reward judge during RL. The reward pathway connects a generated report to its rubric judgments and their aggregation. Appendix C.4 gives the hierarchical aggregation used in the analysis.

G.3 Optimization and Rollout Hyperparameters

The downstream ablation design uses a common starting checkpoint and at least 75 training steps. Exact optimizer and rollout settings are required to compare reward constructions under the same training budget. Table 6 records the configuration used in the reported experiments.

Table 6: Training configuration used in the reported experiments.

Setting	Configuration
Policy initialization	Qwen3-8B-SFT
Rubric generator	GPT5 / Gemini3 variants
Reward judge	Gemini-2.5-Pro
Training hardware	Eight NVIDIA H20 GPUs
Downstream ablation budget	At least 75 steps; same starting checkpoint

G.4 Training Resources and Checkpoint Selection

Training uses eight H20 GPUs. For each run, resource reporting pairs the number of devices with elapsed training time and the evaluated checkpoint. At a constant allocation of eight devices, T hours of training correspond to $8T$ GPU-hours. Rubric-generation and judge API usage are separate costs, measured through calls and tokens.

H Experimental Details

H.1 Benchmarks and Evaluation Sets

H.1.1 Human Preference Alignment

DeepResearch Bench. DeepResearch Bench Du et al. (2026) evaluates open-ended web research reports across diverse domains; here its report comparisons test whether rubric-based scores recover human preferences.

ScholarQA-CS2. ScholarQA-CS2 Bragg et al. (2026) evaluates computer-science literature synthesis with citation-supported answers; here response pairs test preference alignment on scholarly research questions.

H.1.2 Downstream Agent Performance

DeepResearch Bench (DR Bench). DeepResearch Bench Du et al. (2026) evaluates research-report quality through task-specific criteria and reference-based scoring.

DeepResearch Bench II (DR Bench2). DeepResearch Bench II Li et al. (2026b) evaluates information recall, analysis, and presentation using fine-grained binary criteria derived from expert reports.

HealthBench. HealthBench Arora et al. (2025) evaluates responses to healthcare conversations using physician-written, weighted rubrics.

ResearchQA. ResearchQA Yifei et al. (2026) evaluates scholarly question answering across academic fields using questions and answer requirements mined from survey papers.

ScholarQA (SQAv2 in the results table). ScholarQA-CS2 in AstaBench Bragg et al. (2026) evaluates scientific answers through coverage, relevance, and citation quality; the exact ScholarQA release represented by the inherited SQAv2 column requires confirmation before these names can be treated as interchangeable.

H.1.3 Ablation Studies

DeepResearch Bench. DeepResearch Bench Du et al. (2026) supplies general-domain research-report comparisons for the preference-alignment ablations.

ScholarQA-CS2. ScholarQA-CS2 Bragg et al. (2026) supplies scholarly response comparisons for the preference-alignment ablations.

DeepResearch Bench II. DeepResearch Bench II Li et al. (2026b) measures downstream report quality against expert-derived criteria in the agent ablations.

HealthBench. HealthBench Arora et al. (2025) measures healthcare response quality under the alternative reward-construction variants.

ScholarQA (SQAv2). The ScholarQA family Bragg et al. (2026) measures scholarly answer quality in the downstream ablations, subject to the release identification noted above.

H.1.4 Evaluation Direction Quality

ResearchRubrics. ResearchRubrics Sharma et al. (2026) provides expert-authored research prompts and rubrics, which support diagnosis of recovered evaluation scopes against task requirements.

DeepResearch Bench II. DeepResearch Bench II Li et al. (2026b) provides grounded, fine-grained report requirements for examining direction coverage and correctness before rubric generation.

H.1.5 Alignment with Expert-Written Rubrics

HealthBench-Consensus. HealthBench-Consensus Arora et al. (2025) emphasizes healthcare behaviors with strong physician agreement and supplies reference rubrics for the semantic and behavioral alignment comparison.

ResearchRubrics. ResearchRubrics Sharma et al. (2026) supplies expert-written research criteria against which generated rubric content and induced response rankings are compared. The evaluator and candidate responses are held fixed when changing the rubric source.

H.1.6 Individual Rubric Quality

ResearchRubrics. ResearchRubrics Sharma et al. (2026) provides research tasks with expert-defined requirements for assessing individual generated criteria.

DeepResearch Bench II. DeepResearch Bench II Li et al. (2026b) provides report-grounded requirements for checking individual criteria's validity and assessability.

H.1.7 Rubric-Set Analysis, Case Studies, and Training Dynamics

The within- and cross-direction analyses and qualitative cases use the query-level material described in Appendix H.6. The query-level material introduces no additional evaluation set. Training curves use the downstream score definitions below.

H.1.8 Cross-Domain Generalization

SimpleQA. SimpleQA Wei et al. (2024) tests factual accuracy on short questions with verifiable answers.

2WikiMultiHopQA (2Wiki). 2WikiMultiHopQA Ho et al. (2020) tests question answering that requires combining evidence across Wikipedia articles.

WebWalkerQA (WebWalker). WebWalkerQA Wu et al. (2025) tests information seeking through multi-step navigation of web pages.

DeepSearchQA (DSQA). DeepSearchQA Gupta et al. (2026) tests exhaustive web search on questions whose answers may contain multiple distinct items.

H.2 Metric Definitions, Sources, and Adaptations

H.2.1 Human Preference Alignment

Preference accuracy and paired effect size. Following Lv et al. (2026), let $\Delta_i = S(r_i^+) - S(r_i^-)$ be the score difference between preferred and rejected reports. For strictly ordered preference pairs, the cited definitions are

$$\mathrm{Acc} = \frac{100}{N}\sum_{i=1}^{N}\mathbf{1}[\Delta_i > 0], \qquad d = \frac{\mathrm{mean}(\Delta)}{\mathrm{std}(\Delta)}. \tag{27}$$

The factor 100 converts the source accuracy to a percentage. The source uses strictly ordered gold pairs and counts predicted ties as incorrect; its variance divisor is not specified and must be recorded for our scorer. Accuracy measures preference-order recovery; paired d measures standardized score separation. A direct pairwise verdict supplies accuracy but does not supply the scalar score differences needed for d. Gold ties require a separately specified evaluation rule.

H.2.2 Downstream Agent Performance

The following definitions describe the cited evaluation protocols. Their application to each reported row depends on the scorer version and subset recorded with that result (Appendix I.2).

DeepResearch Bench: RACE score. RACE measures report quality relative to a reference report using query-specific dimension and criterion weights Du et al. (2026). For query q and report y, its weighted score and relative score are

$$I_q(y) = \sum_d W_{qd}\sum_k w_{qdk}s_{qdk}(y), \qquad \mathrm{RACE} = \frac{100}{N}\sum_q \frac{I_q(y_q)}{I_q(y_q) + I_q(y_q^{\mathrm{ref}})}. \tag{28}$$

Here d ranges over comprehensiveness, insight/depth, instruction following, and readability; W_{qd} and w_{qdk} are dimension and within-dimension weights, and s_{qdk} is the criterion score. The original overall score uses these weights; an unweighted average of dimension scores is a different aggregation and must be identified if used by a reported source.

DeepResearch Bench II: rubric pass rate. The benchmark measures satisfaction of expert-derived binary criteria Li et al. (2026b). Its per-task score is

$$B_q = \frac{1}{M_q}\sum_{k=1}^{M_q} z_{qk}, \qquad z_{qk} \in \{0, 1\}, \tag{29}$$

where M_q is the number of criteria for query q and z_{qk} indicates whether criterion k is met. Dimension-specific rates restrict the sum and denominator to that dimension. The dataset aggregation must follow the versioned evaluator; averaging the three dimension rates equally does not in general recover the all-criterion score.

HealthBench: weighted rubric score. HealthBench assigns signed point values to physician-written criteria Arora et al. (2025). With criterion weight v_{qk} and satisfaction indicator z_{qk}, the official scorer computes

$$H_q = \frac{\sum_k v_{qk}z_{qk}}{\sum_{k:v_{qk}>0} v_{qk}}, \qquad \mathrm{HB} = 100\ \mathrm{clip}_{[0,1]}\left(\frac{1}{N}\sum_q H_q\right). \tag{30}$$

Negative-weight criteria penalize undesirable behavior; clipping is applied to the dataset mean, not independently to each conversation score.

ResearchQA: rubric coverage. ResearchQA measures how fully an answer addresses survey-derived requirements Yifei et al. (2026). Its released coverage scorer maps five ordered ratings to $a_{qk} \in \{1, 2, 3, 4, 5\}$ and computes

$$\text{ResearchQA} = \frac{100}{N} \sum_{q} \frac{1}{M_q} \sum_{k=1}^{M_q} \frac{a_{qk} - 1}{4}. \tag{31}$$

The endpoints denote no recoverable coverage and sufficiently detailed coverage, respectively; M_q is the number of requirements for query q.

ScholarQA: component scores. AstaBench's ScholarQA-CS2 protocol combines answer coverage, answer relevance, citation precision, and citation recall Bragg et al. (2026):

$$\text{SQA} = \frac{100}{N} \sum_{q} \frac{A_q^{\text{cov}} + A_q^{\text{rel}} + C_q^{\text{prec}} + C_q^{\text{rec}}}{4}. \tag{32}$$

Each component lies in $[0, 1]$: coverage measures the necessary answer content recovered, relevance measures whether answer passages address the question, citation precision measures support from cited sources, and citation recall measures whether answer claims are supported by citations. The SQAv2 column requires a confirmed release and component aggregation before this formula can be assigned to its reported values.

Average across benchmarks. The displayed Avg. is an arithmetic mean of available benchmark scores,

$$\text{Avg}_m = \frac{1}{|\mathcal{B}_m|} \sum_{b \in \mathcal{B}_m} \text{Score}_{m,b}, \tag{33}$$

where $\mathcal{B}_m$ contains the benchmarks reported for model m. This is the table's summary convention, not an additional benchmark metric; rows with missing scores average over different sets.

H.2.3 Ablation Studies

Preference ablations use the same pairwise accuracy definition and benchmark-specific pairs as the full method Lv et al. (2026). Downstream ablations use the DR Bench2 and HealthBench definitions above Li et al. (2026b); Arora et al. (2025), together with the confirmed ScholarQA protocol Bragg et al. (2026). For any of these metrics M, the change caused by removing component c is

$$\Delta M_c = M(\text{full}) - M(\text{without } c). \tag{34}$$

Both terms use the same evaluation set and scorer. For percentage-valued metrics, the difference is in percentage points. This subtraction is a comparison convention, not a new quality metric.

H.2.4 Evaluation Direction Quality

Correctness and unverifiability. Inspired by the evidence-grounded verification in DeepRubric (Figure 12) Zhu et al. (2026), we adapt correctness assessment to individual evaluation directions. The judge receives the query, task context, and the direction's complete description and boundaries, and returns a label $c_q(d) \in \{\text{correct}, \text{incorrect}, \text{insufficient}\}$ with supporting evidence. Coverage, importance, and duplication are assessed separately. Factual claims require verifiable evidence, supplied or retrieved; the absence of expert reference rubrics does not prohibit using factual evidence. Our three-way labeling is an adaptation, distinct from DeepRubric's KEEP/REVISE/DROP revision decisions. For the recovered direction set D_q, define

$$\text{Correctness}_q = \frac{100}{|D_q|} \sum_{d \in D_q} \mathbf{1}[c_q(d) = \text{correct}], \tag{35}$$

$$\text{Unverifiable}_q = \frac{100}{|D_q|} \sum_{d \in D_q} \mathbf{1}[c_q(d) = \text{insufficient}]. \tag{36}$$

Both denominators include all evaluated directions. Insufficient evidence receives no correctness credit and is reported separately rather than relabeled incorrect.

Requirement coverage. RubricBench measures recovery of reference requirements Zhou et al. (2026b). Our direction-level adaptation matches evaluation scope rather than strict criterion equivalence. Let R_q contain the atomic requirements recovered from the expert rubrics, retaining their conditions and positive or negative meaning. Define $m(d, r) = 1$ only when direction d explicitly includes the evaluation target and conditions of requirement r. Then

$$\text{Coverage}_q = \frac{100}{|R_q|} \sum_{r \in R_q} \mathbf{1}[\exists d \in D_q : m(d, r) = 1]. \tag{37}$$

One direction may cover multiple requirements without stating their correct answers in advance. Generic demands for accuracy or comprehensiveness do not cover every target; partial and uncertain matches receive no strict coverage credit. A negative criterion identifies a defect to check, not content the answer should include. This score measures scope coverage of available expert requirements, not completeness over all possible requirements or successful implementation by the generated rubrics.

Task necessity. We adapt RubricBench's instruction-conditioned Intent Necessity assessment Zhou et al. (2026b) to directions, using ratings $n_q(d) \in \{1, \ldots, 5\}$. The judge evaluates whether a direction is needed to answer the query well, considering information, evidence, reasoning, and delivery requirements while allowing alternative valid answers. Occurrence in a reference answer or a particular solution path does not by itself make a direction necessary. We report the mean and rating shares:

$$\text{Necessity}_q = \frac{1}{|D_q|} \sum_{d \in D_q} n_q(d), \qquad B_{q,k} = \frac{100}{|D_q|} \sum_{d \in D_q} \mathbf{1}[n_q(d) = k]. \tag{38}$$

Here $k = 1, \ldots, 5$ ranges from irrelevant or invented requirements to essential requirements; the intervening ratings distinguish tangential, optional, and important requirements. The judge scores necessity alone.

Direction redundancy. RubricRAG's Redundancy@τ measures the proportion of semantically similar pairs Dhole & Agichtein (2026). We replace atomic criteria with complete direction descriptions, including inclusion and exclusion boundaries:

$$\text{Redundancy}_q^D @\tau = \frac{100}{|\mathcal{P}_q^D|} \sum_{(d,d') \in \mathcal{P}_q^D} \mathbf{1}[s(d, d') > \tau], \tag{39}$$

where $\mathcal{P}_q^D$ contains unordered distinct direction pairs. This diagnostic requires no expert reference rubric. The similarity function and threshold are shared across the compared methods.

Accuracy gain from direction weights. Following RRD's comparison of weighting schemes Shen et al. (2026), we compare native and uniform direction weights on the same public preference labels. We fix the query, directions, generated rubrics, item-scoring implementation, candidate response pairs, and judge; only aggregation changes:

$$S_{\text{native}}(y) = \sum_{k=1}^{K} w_k s_k(y), \qquad S_{\text{uniform}}(y) = \frac{1}{K} \sum_{k=1}^{K} s_k(y), \tag{40}$$

$$\Delta \,\text{Acc}_{\text{dir}} = \text{Acc}(S_{\text{native}}) - \text{Acc}(S_{\text{uniform}}). \tag{41}$$

Here w_k are normalized native direction weights and s_k are fixed direction scores. The difference is in percentage points. The optional effect-size comparison is $\Delta d_{\text{dir}} = d(S_{\text{native}}) - d(S_{\text{uniform}})$, using the same paired-score definition as preference alignment. N/A denotes absent native direction weights. This experiment measures weight utility for preference decisions; it does not identify uniquely correct weights. The public preference-label source and its join to each quality-analysis query must be recorded separately from expert rubrics.

H.2.5 Alignment with Expert-Written Rubrics

Semantic alignment: retained set matching. Dhole & Agichtein (2026) compare generated criteria G with expert criteria R through bidirectional best matches. For similarity $s(g, r) \in [0, 1]$, their definitions are

$$P = \frac{1}{|G|} \sum_{g \in G} \max_{r \in R} s(g, r), \quad R_s = \frac{1}{|R|} \sum_{r \in R} \max_{g \in G} s(r, g), \quad F_1 = \frac{2PR_s}{P + R_s}. \tag{42}$$

These are soft matching scores. Strict binary matching, as used by Zhou et al. (2026b), instead counts matched constraints; the two protocols should not be interchanged when interpreting a result.

Behavioral alignment: retained ranking metrics. GenRubric measures whether generated and expert rubrics induce the same response ordering Chen et al. (2026b). For one query, let g_i and e_i be the two scores of candidate response i, with n candidates.

$$\rho_q = \mathrm{Corr}(\mathrm{rank}(g), \mathrm{rank}(e)), \tag{43}$$

$$\tau_{b,q} = \frac{C - D}{\sqrt{(C + D + T_g)(C + D + T_e)}}, \tag{44}$$

$$\mathrm{Top1}_q = \mathbf{1}[G_{\max} \subseteq E_{\max}], \tag{45}$$

$$\mathrm{PairAcc}_q = \binom{n}{2}^{-1} \sum_{i<j} \mathbf{1}[\mathrm{sign}(g_i - g_j) = \mathrm{sign}(e_i - e_j)]. \tag{46}$$

Ranks use average ranks for ties. C and D count concordant and discordant pairs; T_g and T_e count pairs tied only under generated and expert scores, respectively. $G_{\max}$ and $E_{\max}$ contain the maximizing responses under each score. Top-1 consistency allows a subset of the expert top set; pairwise accuracy requires exact agreement on higher, lower, or tied. Table 4 reports these rates as fractions in $[0, 1]$. The source computes each metric per query and generation run, averages runs within a query, and then averages queries, excluding undefined correlations. Our results must also report the number of valid correlations. The number of candidates and repeated runs in our evaluation must be specified separately. These metrics compare rubric-induced judgments, whereas preference accuracy compares against independent human labels.

H.2.6 Individual Rubric Quality

Individual-rubric quality: retained and adapted checks. Correctness checks whether a criterion evaluates its assigned direction and the requested task without incorrect factual or contextual assumptions, adapting RIFT's content-validity and grounding diagnostics (Appendices B.1 and C.1–C.2) Qi et al. (2026). Necessity retains RubricBench's 1–5 instruction-conditioned rating Zhou et al. (2026b), applied to one criterion at a time.

Atomicity follows PReMISE's Atomic check: a criterion receives 1 if it measures one behavioral property and 0 if it bundles distinct properties. Judgeability follows its Operationalizable check: a criterion receives 1 if it can be converted into a concrete binary or ordinal scoring question without the judge inventing additional evaluation rules, and 0 otherwise (Appendices B.1.2 and F.9) Roy et al. (2026). This does not require response-only observability; factual criteria may use the task's evidence.

Discriminability measures the variation of a criterion's binary judgments across candidate responses, following GenRubric Chen et al. (2026b). For criterion k on query q, let $z_{qrk} \in \{0, 1\}$ denote its judgment on candidate response r. The population-standard-deviation form is

$$p_{qk} = \frac{1}{m_q} \sum_{r=1}^{m_q} z_{qrk}, \qquad d_{qk}^{\mathrm{crit}} = \sqrt{p_{qk}(1 - p_{qk})}, \qquad D_q^{\mathrm{crit}} = \frac{1}{|G_q|} \sum_{k \in G_q} d_{qk}^{\mathrm{crit}}. \tag{47}$$

The criterion statistic lies in $[0, 0.5]$, matching the displayed scale. Variation alone does not show that a criterion orders responses by its true quality.

For generated criteria G_q, let $b_c(g)$, $b_a(g)$, and $b_j(g)$ be binary correctness, atomicity, and judgeability judgments, respectively, and let $n(g)$ be the necessity rating. The table's scales are

$$\mathrm{Correctness}_q^{\mathrm{item}} = \frac{100}{|G_q|} \sum_{g \in G_q} b_c(g), \qquad \mathrm{Necessity}_q^{\mathrm{item}} = \frac{1}{|G_q|} \sum_{g \in G_q} n(g), \tag{48}$$

$$\mathrm{Atomicity}_q = \frac{1}{|G_q|} \sum_{g \in G_q} b_a(g), \qquad \mathrm{Judgeability}_q = \frac{1}{|G_q|} \sum_{g \in G_q} b_j(g). \tag{49}$$

Atomicity and judgeability are fractions in $[0, 1]$, correctness is a percentage, and necessity remains on the 1–5 scale. Item judgments use the same response pool and available evidence across methods. Any insufficient-evidence item judgments require separate reporting and a specified denominator, rather than an implicit conversion to incorrect.

H.2.7 WITHIN- AND CROSS-DIRECTION RUBRIC-SET QUALITY

Semantic redundancy: retained statistic, partitioned pairs. RubricRAG's redundancy@τ counts criterion pairs whose similarity exceeds τ Dhole & Agichtein (2026). We apply that statistic to direction pairs and, for rubric sets, separate pairs within the same direction from pairs assigned to different directions:

$$\text{Redundancy}_b\,@\tau = \frac{100}{|\mathcal{P}_b|} \sum_{(i,j)\in\mathcal{P}_b} \mathbf{1}[s(i,j) > \tau], \quad b \in \{\text{within}, \text{cross}\}. \tag{50}$$

Let $a(i)$ denote criterion i's assigned direction. Within each query, $\mathcal{P}_{\text{within}} = \{(i,j) : i < j,\ a(i) = a(j)\}$ and $\mathcal{P}_{\text{cross}} = \{(i,j) : i < j,\ a(i) \neq a(j)\}$. Each numerator and denominator uses the same restricted pair set. For a single-direction panel, restrict the within set further to that direction. No self-pairs are included.

Scoring dependence: retained diagnostic, grouped analysis. RADAR's SymCoupling averages normalized leakage in both directions between two criteria Singh et al. (2026). If $\Delta_{i\to j}$ is the change in criterion j's mean score between high- and low-condition probes targeting criterion i, then

$$\ell_{ij} = \frac{\Delta_{i\to j}}{\Delta_{i\to i}}, \qquad \text{SymCoupling}_{ij} = \tfrac{1}{2}(\ell_{ij} + \ell_{ji}). \tag{51}$$

We separate within- and cross-direction pairs to localize scoring dependence. We additionally report the signed *Directional Coupling Gap*, renaming RADAR's Asymmetry without changing its definition:

$$\text{Gap}_{ij} = \ell_{ij} - \ell_{ji}. \tag{52}$$

A fixed ordering of criterion identifiers determines the sign. A large magnitude indicates unequal influence in the two probe directions; opposite signs must not be averaged away when diagnosing imbalance. A shared legitimate prerequisite can cause coupling, so a nonzero effect is a diagnostic rather than automatic proof of redundant content. For valid pairs in group b, the grouped diagnostic is

$$\text{Coupling}_b = \frac{1}{|\mathcal{V}_b|} \sum_{(i,j)\in\mathcal{V}_b} \text{SymCoupling}_{ij}, \tag{53}$$

where $\mathcal{V}_b$ contains pairs passing the self-effect check in both directions. RADAR normalizes its 0–4 verifier scores to $[0, 1]$, uses a 0.40 self-effect reliability threshold, and computes leakage within task slices before averaging. These source settings must be checked against our probe implementation; they are not inferred from semantic similarity. For direction d on query q, within-direction recall is

$$\text{Recall}_{q,d} = \frac{1}{|R_{q,d}|} \sum_{r\in R_{q,d}} \max_{g\in G_{q,d}} s(r, g), \tag{54}$$

where $G_{q,d}$ contains generated criteria and $R_{q,d}$ contains reference requirements assigned to that direction. This is the direction-conditioned form of RubricRAG recall Dhole & Agichtein (2026); averaging over directions requires the declared direction-weighting convention. The within/cross partition and grouped average are our adaptations of the source diagnostics.

Within-direction weight utility. We adapt RRD's weighting comparison Shen et al. (2026) by holding direction weights w_d, rubric contents, applicability rules, unit assignments, item scores, response pairs, and the judge fixed. For the applicable scoring units $\mathcal{A}_d(y)$ defined in Appendix C.4, replace only their within-direction weights v_{di}:

$$S_{\text{item-native}}(y) = \sum_d w_d \frac{\sum_{i\in\mathcal{A}_d(y)} v_{di} z_{di}(y)}{\sum_{i\in\mathcal{A}_d(y)} v_{di}}, \tag{55}$$

$$S_{\text{item-uniform}}(y) = \sum_d w_d \frac{\sum_{i\in\mathcal{A}_d(y)} z_{di}(y)}{|\mathcal{A}_d(y)|}, \tag{56}$$

$$\Delta\,\text{Acc}_{\text{within}} = \text{Acc}(S_{\text{item-native}}) - \text{Acc}(S_{\text{item-uniform}}). \tag{57}$$

All denominators must be positive. Singleton scoring units give the ordinary per-criterion comparison; any cluster-based implementation must use the same clusters in both conditions. This experiment tests within-direction allocation while preserving each direction's total weight, unlike $\Delta\,\text{Acc}_{\text{dir}}$.

H.2.8 Training Dynamics and Efficiency

Steps to a target score. Following the separation of learning curves and training cost in DR-Tulu Shao et al. (2026), step efficiency compares the first evaluated checkpoint that reaches a shared held-out score θ:

$$t_\theta = \min\{t \in \mathcal{T} : \mathrm{Score}(t) \geq \theta\}. \tag{58}$$

Here $\mathcal{T}$ is the common checkpoint schedule and Score is the benchmark-specific metric defined above. A run that never reaches θ is reported as not reached within its observed budget.

RL GPU-hours and rubric length. GPU-hours quantify accelerator time, as reported by DR-Tulu Shao et al. (2026); mean rubric length measures the text processed by the scorer in our cost analysis:

$$\mathrm{GPUh} = \sum_j g_j h_j, \qquad \overline{L} = \frac{1}{N} \sum_q |\,\mathrm{Tok}(\mathcal{R}_q)|. \tag{59}$$

Interval j uses g_j GPUs for h_j hours, and Tok is the specified tokenizer applied to the serialized rubric set $\mathcal{R}_q$. GPU-hours alone do not include external judge-service cost. The score threshold, tokenizer, and cost accounting boundary require the run records.

H.2.9 Cross-Domain Generalization

SimpleQA: factual accuracy. SimpleQA measures whether a response correctly answers the factual question Wei et al. (2024):

$$\mathrm{Acc}_{\mathrm{QA}} = \frac{100}{N} \sum_q \mathbf{1}[\mathrm{Judge}(y_q, a_q) = \mathrm{correct}], \tag{60}$$

where a_q is the reference answer; incorrect answers and non-answers do not receive credit.

2WikiMultiHopQA: answer matching. The benchmark reports exact match and token-level answer F_1 Ho et al. (2020). For normalized predicted and reference token multisets T_q and U_q,

$$\mathrm{EM}_q = \mathbf{1}[\mathrm{norm}(y_q) = \mathrm{norm}(a_q)], \qquad F_{1,q}^{\mathrm{token}} = \frac{2|T_q \cap U_q|}{|T_q| + |U_q|}. \tag{61}$$

Corpus scores average per-question values and multiply by 100; token intersection counts multiplicity. The reported 2Wiki column needs its actual evaluator identified before assigning it to EM, token F_1, or a judge-based correctness adaptation.

WebWalkerQA: answer accuracy. WebWalkerQA evaluates whether web navigation yields a correct answer Wu et al. (2025). Its correctness rate has the form $\mathrm{Acc}_{\mathrm{QA}}$ above, with the benchmark's answer judgment protocol replacing the SimpleQA judge.

DeepSearchQA: answer-set precision, recall, and F_1. DeepSearchQA evaluates both the correctness and completeness of retrieved answer sets Gupta et al. (2026). For distinct submitted answers A_q and reference answers B_q, after semantic matching,

$$P_q = \frac{|A_q \cap B_q|}{|A_q|}, \quad R_q = \frac{|A_q \cap B_q|}{|B_q|}, \quad F_{1,q}^{\mathrm{set}} = \frac{2P_q R_q}{P_q + R_q}. \tag{62}$$

The benchmark's primary score is $100N^{-1}\sum_q F_{1,q}^{\mathrm{set}}$; its strict fully-correct rate is $100N^{-1}\sum_q \mathbf{1}[A_q = B_q]$. These metrics differ for multi-answer questions, so the DSQA column reports the selected set-level metric. The transfer-table Avg. uses the four displayed task scores with equal weight.

Protocol handling. The equations distinguish source definitions from our direction-level adaptations. Query-level diagnostic formulas above use equal query weight. Gold and predicted ties, empty reference/prediction or pair sets, zero denominators, constant rank vectors, and missing/failed judgments are handled explicitly; undefined observations do not silently become zeros.

H.3 Baseline Inventory and Main-Table Selection

Human preference alignment. The complete comparison in Table 7 includes rubric-free reward models, general-task rubric methods, and LFDR rubric methods. Rubric-free systems include direct GPT and Gemini judges, ArmoRM Wang et al. (2024), Skywork-Reward-V2 Liu et al. (2026a), and DeepSeek-GRM Liu et al. (2025). General-task methods include Auto-Rubric Xie et al. (2026), OpenRS Jia et al. (2026), RubricRM, RubricARM Xu et al. (2026), and Rubric-ARROW. LFDR methods span direct generation, retrieval-augmented generation, search-grounded generation, iterative refinement, and agent–generator co-evolution.
Table 1 retains Tournament-GRPO and QUEST for direct generation, RubricRAG for retrieval, DR-Rubric for search grounding, DeepRubric for iterative refinement, and RubricEM and HOTE for co-evolution. RubricEM and HOTE have the two highest average accuracies among the co-evolving baselines. QUEST complements Tournament-GRPO with stronger ScholarQA-CS2 accuracy and average effect size. The two direct judges and ArmoRM provide rubric-free comparisons. All omitted rows remain in Table 7.

Downstream agents. The main table retains six proprietary systems: Claude-Sonnet Search, Perplexity Deep Research Perplexity AI (2025), Gemini Deep Research Citron (2024), Gemini 3 Pro + Search, GPT-5 + Search, and OpenAI Deep Research OpenAI (2025). It retains the three rubric-free agents with the highest reported averages: FS-Researcher, DualGraph, and WebWeaver Li et al. (2026d). All four static-rubric baselines (DeepResearch Gen., Tournament-GRPO, DeepRubric, and Quest-35B) and all five evolving-rubric baselines (DR Tulu, DR-Rubric, RubricEM, ARISE-RL, and HOTE) remain in the main table. Table 8 additionally includes Perplexity-Sonar (High), Tongyi DeepResearch Tongyi DeepResearch Team et al. (2026), Fathom-DeepResearch, and Web-Thinker Li et al. (2025).

Direction-quality comparators. For Figure 5, DR-Rubric denotes the Stage I *Key Dimensions of Quality* output and its descriptions (Mei et al., 2026); the evaluation concerns this intermediate output, not all requirements distributed across its complete report. Auto-Rubric uses its published high-level dimensions and their subordinate checks (Xie et al., 2026), providing a fixed, query-independent comparator. CARMO denotes our same-input direct-generation adaptation of its prompting approach Gupta et al. (2025): it receives the query and the same high-quality reference responses as Scope2Rubric, but omits explicit/implicit decomposition and cross-response consensus review. These are results of our adapted control, not scores reported by the CARMO paper. Scope2Rubric uses the complete direction-recovery module. All comparisons share queries and diagnostic protocols; a fixed generic direction inventory is not presented as query-specific discovery.

H.4 Evaluation and Judge Configuration

Gemini-2.5-Pro is the common scoring model for rubric-guided preference evaluation and the reward judge during RL. Holding this judge fixed separates the rubric-source comparison from a change in the scoring model. Rubric-free reward models and direct pairwise judges retain their respective evaluation interfaces. Published downstream benchmark scores retain the protocols under which they were reported.
The two S2R variants use GPT- and Gemini-generated rubrics. Downstream agents start from Qwen3-8B-SFT and train on eight H20 GPUs, with the available configuration summarized in Appendix G. Table 2 uses compact backbone names: Tongyi-30B-A3B denotes Tongyi-DeepResearch-30B-A3B, Qwen2.5-7B denotes Qwen2.5-7B-Instruct, and Qwen3.5-35B denotes Qwen3.5-35B-A3B. The full identifiers appear in Table 8.

H.5 Scope2Rubric Implementation and Prompts

The implementation follows three stages whose inputs and outputs are separated by the recovered scope.

Direction recovery. The query and task context first yield explicit directions. Implicit direction discovery uses the same query together with the explicit set. Each reference report is then examined independently for omitted requirements; related candidates are grouped across reports. Coverage, necessity, and priority analyses review the complete candidate set and produce directions with content, boundaries, and normalized weights. The method specifies iteration until the direction set is stable, followed by a deterministic conversion from priority judgments to weights.

Rubric generation. Each generation call is conditioned on a direction's fixed content and boundary. Its output contains necessary judgment points, explicit pass and fail conditions, and unnormalized within-direction weights. Generation-time checks assess atomicity, assessability, and boundary alignment. Set-level review then considers missing judgments and overlap within and across directions.

Hierarchical review and scoring. The method separates semantic duplication from repeated evaluation of a response behavior. Review can merge, remove, narrow, or reassign items. Where distinct necessary judgments share a behavior or prerequisite, the scoring representation records how they are aggregated or made applicable. Each effective unit has one direction assignment, and the direction weights remain fixed. Appendix C.4 gives the conditional aggregation model used by the theoretical analysis.

H.6 Quality Analysis and Case-Study Protocols

Each diagnostic fixes its evaluation unit before aggregating results. Direction analysis operates on the recovered scope before rubric generation. Individual-rubric analysis checks one criterion at a time. Within-direction analysis conditions on a direction boundary, and cross-direction analysis considers pairs with different assignments. The metric definitions in Appendix H.2 specify the respective coverage, semantic overlap, and scoring-dependence views.

Reference requirements and evaluation inputs. ResearchRubrics provides human-authored and reviewed research requirements (Sharma et al., 2026); DeepResearch Bench II derives its criteria from expert reports through extraction, manual revision, and expert review (Li et al., 2026b). The latter is expert-reviewed, not entirely authored from scratch by humans. Reference inventories retain requirement identifiers, conditions, and polarity. Direction scope matching and rubric-level satisfaction are recorded separately, with evidence for each match. Each property is judged independently to avoid using high coverage or high necessity as a substitute for correctness.

Conditional rubric-set analysis. The hierarchical diagnostics examine rubric sets conditional on their assigned direction descriptions and boundaries. They do not assume that all generated directions are correct or complete. Report the direction checks alongside the set results; if a restricted analysis selects only passing directions, state the selection rule and retained counts for every method. This separates a missing direction from missing criteria within an existing direction.
Expert-rubric alignment uses the same candidate responses for both rubric sources. Semantic matching measures agreement between criteria, while behavioral matching compares the resulting response scores or rankings. These comparisons address different questions: two sets can use similar wording yet induce different response rankings, or use different wording while making the same preference decisions.

Aggregation and uncertainty. Compute each diagnostic within a query before averaging queries; report within-direction distributions with query and direction identifiers. Case studies retain this separation at the level of individual records. The complete query and response pair accompany the criteria, and each claimed omission or repeated judgment is linked to a task requirement and response span. A change in the evaluation is calculated on fixed responses; a change in agent behavior is associated with the corresponding training and generation conditions.

I More Results and Analysis

I.1 Complete Human Preference Alignment Results

Table 7 expands the main comparison to the full baseline inventory. It separates rubric-free models, general-task rubric methods, and LFDR rubric methods. Accuracy measures the recovered preference order, while paired d measures standardized score separation; direct verdicts therefore have no reported d when scalar scores are unavailable. Baseline-selection criteria appear in Appendix H.3.

Table 7: Complete comparison of alignment with human preferences on long-form deep research (LFDR) tasks. We report pairwise preference accuracy (Acc.) and paired Cohen's d on DR Bench and ScholarQA-CS2, using Gemini-2.5-Pro as the common scoring model for rubric-guided evaluation. Avg. is the arithmetic mean across the two benchmarks for each metric. Higher values are better; bold and underlining indicate the best and second-best performance in each column, respectively. Dashes denote unavailable values.

Model	Rubric Construction	DR Bench		ScholarQA-CS2		Avg.	
		Acc.↑	d ↑	Acc.↑	d ↑	Acc.↑	d ↑
Rubric-Free Reward Models: Scalar / Generative / Reasoning RMs							
GPT-5.4 (Direct Judge)	N/A	64.09	–	45.05	–	54.57	–
Gemini-3.1-Pro (Direct)	N/A	67.80	–	49.07	–	58.44	–
ArmoRM-Llama3-8B	N/A	44.63	-0.11	44.99	-0.19	44.81	-0.15
Skywork-Reward-V2	N/A	42.08	0.07	42.26	0.01	42.17	0.04
DeepSeek-GRM-27B	N/A	42.62	0.18	49.06	0.19	45.84	0.19
Rubric-Guided RM for General Tasks							
Auto-Rubric	Iterative Refinement	56.18	0.27	43.96	0.18	50.07	0.23
OpenRS	Iterative Refinement	61.70	0.55	46.43	0.21	54.07	0.38
RubricRM	Contrastive Generation	58.49	0.32	44.98	0.13	51.74	0.23
RubricARM	Co-evolving (Generator–Judge)	62.75	0.57	43.28	0.12	53.02	0.34
Rubric-ARROW	Co-evolving (Generator–Judge)	64.26	0.58	47.05	0.10	55.66	0.34
Rubric-Guided RM for Long-Form Deep Research Tasks							
Tournament-GRPO	Direct Generation	72.48	0.48	49.66	0.21	61.07	0.34
RubricRAG	Retrieval-Augmented Generation	44.63	0.19	40.73	0.18	42.68	0.18
QUEST	Direct Generation	71.14	0.58	50.60	0.26	60.87	0.42
DR-Rubric	Search-grounded Generation	72.35	0.53	42.84	0.24	57.60	0.38
DeepRubric	Iterative Refinement	70.28	0.49	45.03	0.21	57.66	0.35
DeepResearch Gen.	Tongyi-DR-30B-A3B	49.3*	0.39	45.07	0.14	56.60	0.27
RubricEM	Co-evolving (Agent–Generator)	74.16	<u>0.66</u>	48.72	0.21	61.44	0.44
HOTE	Co-evolving (Agent–Generator)	67.79	0.37	44.73	0.12	56.26	0.25
DR Tulu	Co-evolving (Agent–Generator)	61.81	0.29	42.38	0.26	52.10	0.28
ARISE-RL	Co-evolving (Agent–Generator)	62.75	0.41	42.92	0.11	52.84	0.26
Ours							
Scope2Rubric (GPT5)	Co-evolving (Agent–Generator)	<u>75.10</u>	0.64	<u>52.84</u>	<u>0.32</u>	<u>63.97</u>	<u>0.48</u>
Scope2Rubric (Gemini3)	Co-evolving (Agent–Generator)	**76.39**	**0.69**	**53.15**	**0.35**	**64.77**	**0.52**

I.2 Complete Downstream Agent Results

Table 8 retains the full downstream comparison, including systems omitted from the compact main table. The source and subset markers identify the provenance of individual cells. Because different rows report different subsets of benchmarks, their averages summarize the available entries rather than a common complete evaluation set. Model-specific conditions follow the table.

Result sources and evaluation conditions. Stars identify scores matched to published reports after rounding. Proprietary-system, DR Tulu, and Tongyi scores are reported in Table 1 of Shao et al. (2026); DeepRubric and RubricEM scores match their respective Table 1 (Zhu et al., 2026; Li et al., 2026a). Other starred scores match the corresponding method or benchmark papers. DR Bench2 scores for Perplexity, Gemini Deep Research, OpenAI Deep Research, and Tongyi match Table 3 of Li et al. (2026b); the Gemini and OpenAI entries refer to Gemini-2.5-Pro and OpenAI-

Table 8: Complete downstream comparison across five LFDR benchmarks. * denotes scores reported in prior work; † denotes evaluation on a 100-sample subset. Our models start from Qwen3-8B-SFT. Avg. averages available scores. A dash indicates that the corresponding benchmark score is not reported in the cited source. Bold and underlined values mark the best and second-best open-model scores.

Model	Backbone	DR Bench	DR Bench2	Health Bench	Research QA	SQAv2	Avg.
Proprietary DeepResearch							
Claude-Sonnet Search	Claude Sonnet	34.5*	–	–	64.3*	–	49.4
Perplexity-Sonar (High)	Proprietary	40.7*	–	–	69.1*	–	54.9
Perplexity Deep Research	Proprietary	42.3*	38.6*	–	75.3*	67.3*	55.9
Gemini Deep Research	Proprietary	48.8*	42.0*	–	68.5*	–	53.1
OpenAI Deep Research	o3/o4	46.9*	45.4*	53.8*,†	79.2*,†	79.6*	61.0
Gemini 3 Pro + Search	Gemini 3 Pro	46.3*	34.6	38.0*	74.3*	69.8*	52.6
GPT-5 + Search	GPT-5	50.7*	–	59.5*,†	78.2*,†	74.8*	65.8
Rubric-Free DeepResearch							
Tongyi DeepResearch	Qwen3-30B-A3B	40.6*	29.9*	46.2*	66.7*	46.5*	46.0
Fathom-DeepResearch	Qwen3-4B	45.5*	33.4	48.6	70.5	70.5	53.7
FS-Researcher	GPT-5	52.8*	40.7	59.3	79.9	79.8	62.5
DualGraph	GPT-5 / GPT-4.1	<u>53.1</u>*	41.5*	60.0	78.8	81.1	62.9
WebWeaver	GPT-4.1	45.2*	36.6*	53.3	76.5	78.8	58.1
WebThinker-DPO report	Qwen2.5-32B	40.6*	30.3	39.4*	74.2*	46.7*	46.2
RL with Static Rubric Rewards for DeepResearch							
DeepResearch Gen.	Tongyi-DR-30B-A3B	49.3*	40.4	47.9	78.5	85.4	60.3
Tournament-GRPO	Qwen2.5-7B-Instruct	50.2*	38.3	53.6	79.4	86.1	61.5
DeepRubric	Qwen3-8B	43.6*	33.5	45.1	75.2*	86.0*	56.7
Quest-35B	Qwen3.5-35B-A3B	48.2*	42.6	57.2	77.1	85.6	62.1
RL with Evolving Rubric Rewards for DeepResearch							
DR Tulu	Qwen3-8B	45.4*	37.0	52.8*	75.7*	88.3*	59.8
DR-Rubric (BS-3)	Qwen3-8B	39.5*	31.6	46.9	72.4*	84.1	54.9
RubricEM	Qwen3-8B	47.8*	38.5	49.3*	74.5*	85.8	59.2
ARISE-RL	Qwen3-8B	46.1	35.1	52.7	74.8	82.4	58.2
HOTE	Qwen3-8B	45.9*	32.4	54.4*	76.9*	80.5	58.0
Ours							
Qwen3-8B-SFT	Qwen3-8B	38.4	29.6	35.6	59.3	60.9	44.8
Scope2Rubric-8B (GPT5)	Qwen3-8B	52.5	<u>47.2</u>	<u>62.1</u>	**82.8**	<u>89.7</u>	<u>66.9</u>
Scope2Rubric-8B (Gemini3)	Qwen3-8B	**54.1**	**48.4**	**65.2**	<u>80.5</u>	**90.4**	**67.7**

GPT-o3, respectively. Daggers identify the 100-sample HealthBench and ResearchQA evaluations of OpenAI Deep Research and GPT-5 + Search. Averages use the displayed entries.

Model configurations. DualGraph uses GPT-5 for DR Bench and GPT-4.1 for DR Bench2; its row summarizes these two configurations. WebWeaver's GPT-4.1 scores are reported in DualGraph's Tables 1 and 3. DeepResearch Gen. uses Tongyi-DeepResearch-30B-A3B (Tongyi-DR-30B-A3B in the table) with MaMs and a Qwen3-30B-A3B rubric generator trained by RL (Lv et al., 2026). DR-Rubric denotes the BS-3 variant (Mei et al., 2026). WebThinker-DPO report uses Qwen2.5-32B, as identified in Table 1 of Zhu et al. (2026).

Short-form results. The starred scores in Table 5 match Table 2 of Li et al. (2026a): DR Tulu uses the 1,900-step checkpoint, and RubricEM uses the 1,400-step checkpoint. Their averages are reproduced as reported, including DR Tulu's 49.0; averaging its rounded entries gives 48.9.

I.3 Individual Rubric Quality

Table 9 compares individual rubrics across five quality dimensions. Human rubrics score highest on every dimension, followed by Scope2Rubric, Tournament-GRPO, and RubricEM on both datasets.

Table 9: Individual rubric quality. Higher values are better. Bold marks the best result and underlining the second-best result within each dataset.

Model	Correctness (%) ↑	Necessity (1–5) ↑	Atomicity (0–1) ↑	Discriminability (0–0.5) ↑	Judgeability (0–1) ↑
ResearchRubrics					
Human rubrics	**77.93**	**4.35**	**0.85**	**0.42**	**0.82**
Tournament-GRPO	50.67	3.90	0.75	0.36	0.70
RubricEM	47.45	3.78	0.70	0.30	0.67
Scope2Rubric (Gemini-3)	<u>60.88</u>	<u>4.25</u>	<u>0.81</u>	<u>0.40</u>	<u>0.75</u>
DeepResearch Bench II					
Human rubrics	**52.37**	**4.20**	**0.80**	**0.40**	**0.78**
Tournament-GRPO	38.62	3.67	0.73	0.32	0.68
RubricEM	36.48	3.22	0.69	0.28	0.64
Scope2Rubric (Gemini-3)	<u>49.04</u>	<u>4.06</u>	<u>0.77</u>	<u>0.37</u>	<u>0.74</u>

I.4 Additional Direction and Rubric-Set Analyses

Direction-level diagnostics. Figure 5 summarizes directions before rubric generation. It distinguishes the correctness and necessity of included directions from coverage of reference requirements and overlap between directions. The native-versus-uniform weighting comparison holds the direction scores fixed, isolating the change in the aggregation weights. Thus, a coverage change and an accuracy-gain change describe different properties of the recovered scope. The paired ResearchRubrics and DR Bench2 panels are shown together in the main-text Figure 5; the panel-level PDFs remain available as separate source assets for the additional two positions in the four-panel row.

Unverifiability, necessity, and scope coverage. The extended direction results report unverifiable rates alongside correctness and show the five necessity-rating shares. A query-level heatmap aligns expert requirement identifiers with generated directions, distinguishing strict scope matches from partial or uncertain matches. Each case links its correctness verdict to the relevant evidence and boundary. These records complement the aggregate direction table without treating unavailable judgments as errors.

Expert alignment and set quality. Table 4 separates criterion matching from response-order agreement. For example, on ResearchRubrics, the displayed S2R semantic F1 is 0.53, while top-1 consistency is 0.38. These metrics have different denominators and should be interpreted within their own definitions. Improved criterion overlap can coexist with disagreement about which response is best. The within- and cross-direction diagnostics further distinguish missing judgment points from repeated evaluation.

Within- and cross-direction presentation. The six-panel analysis separates (a) within-direction reference recall, (b) within-direction semantic redundancy, (c) within-direction SymCoupling versus Directional Coupling Gap, (d) within-direction weighting accuracy gain, (e) cross-direction semantic redundancy, and (f) cross-direction SymCoupling versus Directional Coupling Gap. Panels (c) and (f) share axes and method colors. Redundancy and weighting-gain panels report uncertainty; recall is shown as a distribution. A complementary recall–redundancy scatter in the appendix links both measurements for the same query and direction, revealing whether reduced repetition comes at the expense of reference coverage.

I.5 Additional Ablation Results

The ablation design separates direction recovery, decomposition, and hierarchical review. Removing directions also replaces the direction-based review with flat review. Removing hierarchy retains directions but uses flat review. These variants address different questions: the former changes both the intermediate representation and its review structure, while the latter targets review structure conditional on having directions.

Preference-alignment and fixed-endpoint downstream comparisons use the four variants in Table 3. Per-task training curves and run-level uncertainty are still needed to determine whether the relative component effects agree between preference evaluation and policy training.

I.6 Full Training Curves and Cost Analysis

Training dynamics connect the in-loop reward to held-out performance. For a common evaluation schedule, step efficiency compares the number of optimization steps required to reach the same benchmark score. GPU-hours additionally account for the resources and time used per step. A checkpoint's training reward and downstream benchmark score are reported separately because they can use different judges and criteria.

Rubric-generation and scoring costs complement RL compute. Calls and tokens characterize the generator and judge workload, while the average rubric-set length describes the evaluation material passed to the judge. Tool calls, response length, and citation-related behavior contextualize changes in the agent's trajectories.

I.7 Cross-Domain Generalization

Table 5 tests transfer to short-form factual and search-intensive tasks. The displayed S2R average is 75.5, compared with 73.5 for RubricEM and 71.4 for Tournament-GRPO. Relative to RubricEM, the gains are 1.6 points on 2Wiki, 6.7 on WebWalker, and 2.1 on DSQA, with a 2.5-point decrease on SimpleQA. The average therefore summarizes task-dependent changes rather than a uniform improvement.

The starred baseline scores retain the checkpoints and reporting conventions described in Appendix I.2. This transfer comparison concerns the trained agent's task performance; it does not directly measure how accurately the recovered scope matches the requirements of each short-form query.

J Analysis on Searched Tools and Domain Distributions

Tool use and retrieved web domains describe how an agent gathers evidence. They complement the quality scores by locating differences in information acquisition across tasks. Web domains identify source websites; task domains identify categories of research questions.

J.1 Tool Usage across Tasks

For an evaluated query set $\mathcal{Q}$, let n_{qt} be the number of attempted calls to tool type t on query q. The pooled call share is

$$p_t = \frac{\sum_{q \in \mathcal{Q}} n_{qt}}{\sum_{q \in \mathcal{Q}} \sum_{t'} n_{qt'}}, \tag{63}$$

when the denominator is positive. Mean calls per query, $|\mathcal{Q}|^{-1} \sum_q n_{qt}$, provide the corresponding absolute usage level. Reporting both prevents a change in relative share from being mistaken for an increase in the number of calls. Failed calls are retained among attempts and identified separately. Comparisons use the same tool taxonomy and specify the policy checkpoint, query set, and call budget.

J.2 Web-Domain Distributions

Retrieved URLs can be grouped by their source domain to characterize the evidence pool. Repeated retrieval of the same URL and retrieval of multiple pages from one website affect the distribution differently. The analysis therefore identifies whether its unit is a retrieval event, a unique URL per query, or a distinct source domain per query, and records the normalization of subdomains and redirects.

The resulting distribution describes source selection. A high share from one site establishes concentration of retrieval, while claims about evidence quality require checking the pages and their use in the report.

K Discussion and Future Work

K.1 Scope-First Evaluation for LFDR

An explicit evaluation scope makes the connection between a query and its reward inspectable at two levels. Directions identify the judgments that matter to the task, and rubric sets turn those judgments into response-level checks. This separation allows missing requirements to be diagnosed before item refinement and keeps direction priorities visible as the rubric granularity changes.

The theoretical analysis clarifies why these levels matter. A rubric set cannot distinguish responses that it maps to the same score vector; evaluation error can arise from both the allocation of judgment weight and the estimated satisfaction of those judgments. Direction-wise normalization preserves a fixed weight budget under its stated conditions. The empirical analyses then address complementary questions: whether the directions match task requirements, whether the rubrics agree with expert criteria, and whether their rewards support the agent.

K.2 Limitations

Scope recovery depends on the task interpretation and the available reference reports. References can expose an omission, but shared omissions or unsupported details can also persist across reports. Task-grounded acceptance checks reduce reliance on any single report while leaving the quality of those checks consequential for the final scope. Once a scope is fixed, item-level review cannot repair a missing direction without reopening the scope-level decision.

Scoring introduces a further source of error. A valid criterion can be misjudged, and shared evidence or prerequisites can create dependence between otherwise distinct criteria. The theoretical results concern the stated aggregation model; their connection to a particular run depends on its actual scoring-unit and applicability rules. Empirical coverage is measured against available reference requirements, whose completeness determines what the diagnostic can reveal.

Finally, direction recovery and hierarchical review require additional generation and judging work. Their practical value depends on the resulting performance–cost trade-off and on how well the evaluation transfers across task and judge distributions. Comparisons involving published scores also retain differences in model configuration, evaluation subsets, and available benchmark coverage.

K.3 Future Work

One direction is to make scope revision explicit: a failed case can identify whether repair belongs at the direction level or within the existing rubric set. Such a mechanism would need to record changes to content, boundaries, and priorities so that revisions remain distinguishable from item refinement.

A second direction is to study the marginal value of reference reports and review passes under a fixed construction budget. The relevant question is how much additional task coverage or preference discrimination each stage provides relative to its cost. A third direction is to examine judge transfer and dependency-aware scoring, connecting semantic overlap to controlled changes in response scores. These studies would further link the recovered scope to the reliability and efficiency of rubric rewards.

L Qualitative Case Study

Figure 9 makes the construction contrast explicit for one ResearchQA query. The top row fixes its requirement set; the two columns then show how the rubric-first control and Scope2Rubric construct their evaluation objects. Each Scope2Rubric direction records its scope attributes, and each rubric records a title, pass condition, and fail condition before the overall set is scored on the same response pair.

Requirement set. **R1** feature-level semantics; **R2** level–attribute alignment; **R3** level–task–performance explanation; **R4** evidence and applicability boundaries.

Rubric-first control
flat rubric construction

Core contrast. Direct rubric expansion does not first fix a query-specific semantic and boundary scope, so salient off-task content can receive credit. Scope2Rubric fixes that scope before rubric generation, so B's valence/arousal detour is excluded from credit.

Failure mode. Without a direction boundary, local checks can reward the response's salient but task-irrelevant detour.

Decision under the flat set. A: 81.40; B: 84.96; prediction: B (wrong)

Scope2Rubric
scope-first construction

Evaluation scope. Four directions use $d_i = (\mathcal{U}_i, e_i, \phi_i, w_i)$ and $e_i = (c_i, b_i)$. The artifact records c_i (what is evaluated and its quality meaning) as `direction_semantics.description`, and b_i (the included/excluded range) as `coverage_and_boundary`.

D1 *content.semantic* c_i*:* effects of feature levels on facial-attribute-analysis performance; *content.boundary* b_i*:* include CNN-level effects for this task, exclude generic layer descriptions and other tasks. *structure* ϕ_i*:* direct relation; *weight* w_i*:* .2167.
r1 T: directly relate feature level to performance; *P:* connects level, facial attribute analysis, and performance; *F:* only defines levels.

D2 *content.semantic* c_i*:* meanings of shallow, middle, and deep levels and their support for facial attributes; *content.boundary* b_i*:* include level-specific semantics and attribute dependence, exclude interchangeable levels. *structure* ϕ_i*:* alignment relation; *weight* w_i*:* .2167.
r1 T: distinguish shallow, middle, and deep semantics; *P:* explains each level's information; *F:* treats levels as interchangeable.
r2 T: align attribute types with feature-level information; *P:* matches attributes to levels; *F:* assigns one level to all.

D3 *content.semantic* c_i*:* why a feature level matches a task and yields a performance difference; *content.boundary* b_i*:* include conditional level–task–performance explanations, exclude universal level rankings. *structure* ϕ_i*:* explanatory chain; *weight* w_i*:* .2167.
r1 T: continuous level–task–performance chain; *P:* connects representation, task match, and outcome; *F:* gives isolated definitions.
r2 T: conditional synthesis; *P:* states when a level is useful; *F:* claims one level is universally best.

D4 *content.semantic* c_i*:* evidence and conditions supporting the explanation; *content.boundary* b_i*:* include evidence and task/data applicability limits, exclude unsupported universal claims. *structure* ϕ_i*:* qualification; *weight* w_i*:* .3500.
r1 T: minimal empirical support; *P:* anchors the explanation in evidence; *F:* makes unsupported claims.
r2 T: applicability conditions; *P:* marks task/data limits; *F:* presents universal optimality.

Overall rubric set. 4 directions, 7 rubrics.
A: 100.00 B: 93.74 prediction: A (human-aligned)

Same response pair. **A** stays on CNN feature semantics and facial attribute analysis; **B** shifts to valence/arousal prediction and `CNN-3RNN-2nd-pool_last-pool_fc`.

Human preference: A **Scope2Rubric: A (correct)** **Rubric-first: B (incorrect)**

Figure 9: **Evaluation Scope and Rubric Comparison** for one ResearchQA query. Both columns use the same requirements and response pair. Scope2Rubric exposes each direction as *content=(semantic,boundary), structure, and weight* before refining it into direction-specific rubrics with title (T), pass (P), and fail (F) conditions.

Rubric Set Comparison. The left and right columns trace the same query through the two construction orders. The flat control's 39 items include local checks but do not expose a query-specific direction boundary. Scope2Rubric makes the ownership boundary explicit: each rubric is generated inside a direction's content *(semantic, boundary)* and structure before the overall set is assembled.