RUBRICS BEYOND EVALUATION: EXPLOITING RUBRIC STRUCTURE IN REINFORCEMENT LEARNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Rubrics are widely used to evaluate open-ended question answering by breaking overall quality into explicit criteria (Kim et al., 2024; Arora et al., 2025). These explicit criteria make rubrics a rich source of learning guidance, yet their potential for guiding the training process has been largely overlooked. Existing methods use rubrics mainly as direct feedback signals, without explicitly modeling the capabilities behind their criteria (Gunjal et al., 2025; Li et al., 2026). This leaves their capability structure largely unused in guiding the training process. We introduce Rubric-Structured Reinforcement Learning (RSRL), a RL framework that unlocks the full potential of rubrics by exploiting their structure throughout the training process, rather than treating them simply as direct reward signals. At its core, RSRL builds on three key designes: (1) ***Rubric-to-Capability Structuring*** first recovers the capability structure hidden in rubric criteria, (2) ***Self-Sufficient Capability Learning*** then uses this structure to bootstrap underdeveloped capabilities into effective learning, (3) ***Capability-Residual Credit Routing*** finally preserves capability-specific gains during optimization. Experiments show that RSRL consistently improves both overall performance and underdeveloped capabilities, boosting downstream scientific question-answering performance by up to 7.43 over the initial policy and 7.60 over non-rubric structured RL baselines, showing that reinforcement learning can substantially benefit from the capability structure encoded in rubrics.

1 INTRODUCTION

Rubrics (Kim et al., 2024) have become increasingly popular for evaluating complex model outputs across a wide range of open-ended tasks (Arora et al., 2025; Yifei et al., 2025), as they provide a structured way to break overall quality into explicit criteria. These criteria can capture aspects such as correctness, completeness, evidence use, reasoning quality, and clarity. More importantly, these criteria are not merely scoring rules, they reveal the capabilities required to produce high-quality responses, making rubrics ***a rich source of guidance for model learning***. However, this potential has barely been explored. Most existing work (Kim et al., 2024; Arora et al., 2025) uses rubrics only for evaluation, and the few attempts (Li et al., 2026; Gunjal et al., 2025; Shao et al., 2026) that incorporate them into reinforcement learning, they largely treat individual criteria as direct supervision or reward signals. As a result, the capability structure encoded across rubric criteria ***remains largely unused*** during learning.

We argue that rubrics should not be treated merely as collections of reward criteria, but as structured descriptions of the capabilities required to solve a task. The key is to preserve and exploit this structure throughout reinforcement learning, rather than collapsing it into a single aggregate training signal. This shifts rubric-based learning ***from criterion-level optimization to capability-level development***: rubric criteria reveal what capabilities the policy lacks, and the capability structure can further guide how learning resources are allocated during training. Based on this view, we formulate **Rubric-Structured Reinforcement Learning (RSRL)**, which turns rubrics from an evaluation tool into a structure that actively shapes the learning process.

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.

Three key challenges arise when bringing rubric structure into reinforcement learning: **(C1)** ***What capability is underdeveloped?*** Rubric criteria are defined at the question level, while the weaknesses they reveal may reflect shared capabilities across many questions, making it difficult to recover these shared capabilities and identify which ones remain weak; **(C2)** ***How can weak capabilities be effectively explored?*** Once a weak capability is identified, natural rollouts may still provide too little useful variation for the policy to discover better behaviors, making it difficult to achieve effective early-stage improvement. **(C3)** ***How can capability gains be preserved?*** Even when better capability-specific behaviors are found, their signals can be diluted once multiple rubric criteria are aggregated during optimization, causing the improvement to have little effect on the final policy update.

To address these challenges, our RSRL carries rubric structure through the full learning process rather than using it only at the reward level. (For **C1**) we introduce **Rubric-to-Capability Structuring** to recover the capability structure hidden in rubric criteria. Specifically, we construct Hierarchical Capability Channels to organize fragmented criteria into shared capabilities and track which capabilities remain underdeveloped across questions. **(For C2), Self-Sufficient Capability Learning** is designed to bootstrap underdeveloped capabilities when the current policy lacks useful learning behaviors, while allowing them to transition toward independent learning as they improve. To realize this idea, we develop a Self-Sufficiency Gating Module that introduces capability-enhanced rollouts through capability-specific prompting when additional support is needed, and adaptively withdraws them once the capability becomes sufficiently established. (For **C3**), we further introduce **Capability-Residual Credit Routing** to preserve capability-specific gains during aggregate optimization. In practice, we formulate a Capability Residual Projection Policy to separate the target capability's residual improvement from the aggregate advantage and carry this signal into the final policy update. Experiments on ResearchQA (Yifei et al., 2025), closed-book SQAv2 (Bragg et al., 2026), and GPQA-Diamond (Rein et al., 2024) show that RSRL improves downstream scientific question-answering performance by up to 11.52% over the initial policy and 11.81% over non-rubric RL baselines. It also achieves the best final worst-capability and overall response scores on a fixed 128-question development probe, while preserving gains in strong capabilities. These results demonstrate the value of exploiting rubric-encoded capability structure throughout reinforcement learning.

Our main contributions are as follows:

- **A systematic formulation of rubric-structured reinforcement learning.** We are the first to deeply examine how rubric structure can be exploited beyond evaluation and direct reward supervision, revealing its role in capability modeling, learning allocation, and policy optimization. This establishes RSRL as a general framework for integrating rubric structure throughout reinforcement learning.
- **A structured RL framework built around capability development.** We instantiate this view with RSRL through three coordinated designs: *Rubric-to-Capability Structuring* recovers shared capability structure from rubric criteria, *Self-Sufficient Capability Learning* enables efficient improvement of underdeveloped capabilities, and *Capability-Residual Credit Routing* preserves capability-specific gains during policy optimization.

2 Related Work

Rubric-based supervision and exploration. Rubrics as Rewards uses instance-specific criteria as RL feedback (Gunjal et al., 2025); RubricHub scales rubric construction and rubric-based post-training (Li et al., 2026). DR Tulu evolves criteria with the policy using newly retrieved evidence and contrasting responses (Shao et al., 2026). RuscaRL guides exploration with rubric scaffolds that decay over training (Zhou et al., 2025). RISE-RL targets repeatedly missed criteria and learns from reward-filtered guided trajectories through an auxiliary objective under the original prompt (Hou et al., 2026). CRAFT builds a hierarchical capability tree from rubric criteria to guide targeted supervised fine-tuning (Gupta et al., 2026). The RL methods above organize reinforcement around question-specific criteria, without a shared capability diagnosis that jointly coordinates online exploration and response-level credit across questions.

Adaptive allocation and reward balancing. Automated curricula allocate training exposure using learning progress (Graves et al., 2017). For multi-reward optimization, GDPO normalizes reward dimensions separately (Liu et al., 2026b), GD^{2}PO filters conflicting rollouts and reweights queries by reward agreement (Liu et al., 2026a), and SMOPD distills reward-specialized teachers into one policy (Wang et al., 2026). These methods manage training exposure and reward trade-offs, but do not specify how persistent capability deficits should jointly determine exploration support and response-level credit.

Controlled generation and sampling correction. PREADD contrasts distributions with and without a control prefix (Pei et al., 2023), while contrastive decoding compares stronger and weaker models (Li et al., 2023). Multiple importance sampling provides density-based weights for combining samples from different proposals through the balance heuristic (Veach & Guibas, 1995). These tools support generation control and sampling correction, but leave unspecified when a weak capability needs guidance and when that guidance should be withdrawn during learning.

Together, these gaps motivate a learning process organized around evolving capability needs. RSRL uses rubric structure to connect shared capability diagnosis, selective exploration, and capability-specific policy updates, making the policy's diagnosed capability state govern the provision and withdrawal of exploration support.

3 Problem Motivation

Rubric-based RL needs sampled responses that provide useful variation on weak capabilities. When unguided rollouts rarely satisfy a requirement, rubric feedback can identify the gap without supplying successful behaviors to learn from. Reweighting scores alone cannot introduce behaviors absent from the sampled group. Self-Sufficient Capability Learning uses capability-guided proposals when diagnostics identify insufficient support and a sufficiently large deficit, then withdraws guidance when its activation conditions no longer hold. Guided responses are learned under the original question context, with sampling weights accounting for the proposal shift.

RSRL connects this exploration process to diagnosis and optimization through a shared capability target. Rubric-to-Capability Structuring groups question-local criteria into cross-question channels and combines current evidence with history. Capability-Residual Credit Routing carries target-specific distinctions into response-level advantages. Together, these stages coordinate which capability receives exploration support and how its feedback affects policy updates.

4 Rubric-Structured Reinforcement Learning

RSRL uses rubric structure to identify weak capabilities, guide exploration, and route capability-specific signals into policy updates under the original question (Figure 1).

4.1 Rubric-to-Capability Structuring

Rubric-to-Capability Structuring uses **Hierarchical Capability Channels** to link question-specific rubric requirements with recurring domain capabilities.

Constructing capability channels. For a question q, let x_q denote its original context, $\mathcal{R}_q$ its rubric criteria, and $\mathcal{C}_q$ its applicable local capability channels. Each source criterion is counted once and assigned to one local channel. A domain codebook $\mathcal{Z}$ links channels across questions through

$$m_q : \mathcal{C}_q \to \mathcal{Z} \cup \{\perp\}, \tag{1}$$

where $\perp$ denotes no reliable match. An LLM proposes the definitions, and an LLM judge assesses their correspondence. Definitions and mappings are frozen before training. Unmatched channels retain local diagnosis and training credit but do not update domain state (Appendix A).

Estimating capability performance. Before update s, we sample unguided diagnostics from $\pi_{\theta_{s-1}}(\cdot \mid x_q)$ separately from optimization rollouts. Criterion scores lie in $[0, 1]$, with higher values

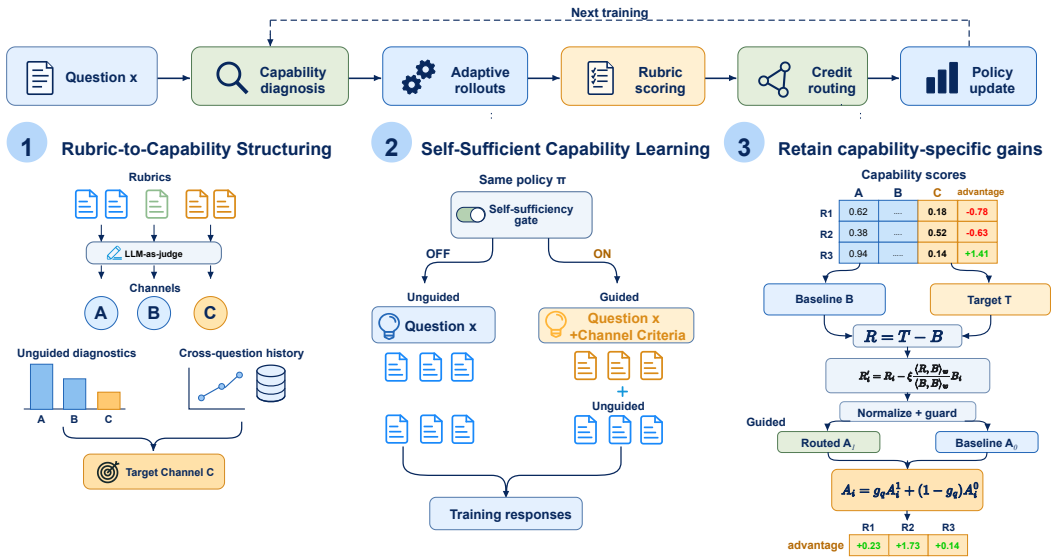

Figure 1: Overview of RSRL. Rubric-to-Capability Structuring identifies targets from local and cross-question evidence. Self-Sufficient Capability Learning gates guided exploration. Capability-Residual Credit Routing adjusts response credit, with a guard selecting the routed candidate or baseline. Sampling weights account for the proposal shift; updates use the original question.

better. We average valid observations, including zeros, per criterion and combine them with fixed within-channel weights to obtain $S_{q,c} \in [0,1]$.

For a mapped channel, we pool current observations with cross-question history whenever both estimates are available:

$$\widetilde{S}_{q,c} = \nu_{q,c} S_{q,c} + (1-\nu_{q,c}) U_{m_q(c),s-1}, \qquad \nu_{q,c} = \frac{n_{q,c}}{n_{q,c}+\kappa}. \tag{2}$$

Here $U_{z,s-1}$ is the historical domain estimate, $n_{q,c}$ counts distinct responses with valid channel evidence, and $\kappa > 0$ controls pooling. We use either estimate directly if the other is unavailable; channels lacking both are excluded from target selection.

Selecting the target capability. Channels with available estimates form $\mathcal{C}_q^{\text{diag}} \subseteq \mathcal{C}_q$. A fixed performance threshold $\tau \in (0,1]$ defines their normalized deficit:

$$h_{q,c} = \max\left(0, \frac{\tau - \widetilde{S}_{q,c}}{\tau}\right). \tag{3}$$

If $\mathcal{C}_q^{\text{diag}}$ is empty, targeted exploration and residual routing are disabled for that question. Otherwise, we select the channel with the largest estimated deficit:

$$c_q^* = \arg\max_{c \in \mathcal{C}_q^{\text{diag}}} h_{q,c}, \qquad h_q^* = h_{q,c_q^*}. \tag{4}$$

Stable channel identifiers break ties. If $h_q^* = 0$, both exploration and routing coefficients are zero. Reliability $\chi_q \in [0,1]$ combines local evidence, domain-prior confidence, and mapping confidence for routing (Appendix F).

Maintaining the domain state. We initialize domain estimates with unguided diagnostics of the starting policy. At each update, observations sharing a domain capability are merged within questions and averaged equally across questions to obtain $\bar{U}_{z,s}$. After the optimizer step,

$$U_{z,s} = \beta_D U_{z,s-1} + (1-\beta_D)\bar{U}_{z,s}, \tag{5}$$

where $\beta_D \in [0,1)$ controls historical retention. The selected channel and deficit guide exploration and credit allocation in Sections 4.2–4.3.

4.2 SELF-SUFFICIENT CAPABILITY LEARNING

Self-Sufficient Capability Learning uses a **Self-Sufficiency Gating Module** to activate capability-guided sampling while retaining the original question for policy updates.

Deciding when guidance is needed. Guidance requires sufficient diagnostic evidence, a deficit above the activation threshold, and either uniformly minimal scores (**Zero**) or varying scores with few positive responses (**Low**). Other cases use unguided sampling. Reassessment before each update withdraws guidance whenever an activation condition fails, tying exploration support to the policy's current diagnostic state rather than a fixed training schedule. These are observed score patterns, not probability-support claims (Appendix B.2).

Constructing guided proposals. At a shared response prefix $y_{<t}$, we evaluate three distributions from the same pre-update policy: $p_{0,t}$ conditions on x_q; $p_{c,t}$ adds the target capability description and training criteria; and $p_{n,t}$ adds nondirectional guidance. This target–control comparison provides a reference for distinguishing capability cues from the effects of extra prompting, adapting prefix-based control (Pei et al., 2023).

Using the full softmax distributions, we form the target–control contrast

$$u_t(v) = \log p_{c,t}(v) - \log p_{n,t}(v), \tag{6}$$

where v denotes a candidate token. We standardize this contrast under the unguided distribution and clip its magnitude:

$$\delta_t(v) = \mathrm{clip}\left(\frac{u_t(v) - \mathbb{E}_{v' \sim p_{0,t}}[u_t(v')]}{\sqrt{\mathrm{Var}_{v' \sim p_{0,t}}[u_t(v')] + \varepsilon}}, -b, b \right). \tag{7}$$

Here, $b > 0$ bounds the standardized contrast, and $\varepsilon > 0$ stabilizes the variance normalization. We apply the resulting perturbation to the unguided distribution:

$$q_{\eta,t}(v) = \frac{p_{0,t}(v) \exp(\eta_q \delta_t(v))}{\sum_{v'} p_{0,t}(v') \exp(\eta_q \delta_t(v'))}. \tag{8}$$

The strength $\eta_q \geq 0$ increases with h_q^* up to a cap, with a smaller multiplier for **Low**. At $\eta_q = 0$, the proposal is $p_{0,t}$. Appendix B.2 specifies the schedule and prompt content.

Learning under the original question. An active gate allocates optimization responses equally between guided and unguided sampling; otherwise, all are unguided.

Guided sampling can change the response distribution. We account for this shift using sequence-level importance weights based on the balance heuristic of Veach & Guibas (1995). Let $P_0(y \mid x_q)$ denote the pre-update unguided sequence distribution. The guided sequence distribution is

$$Q_\eta(y \mid x_q) = \prod_{t=1}^{|y|} q_{\eta,t}(y_t), \tag{9}$$

where each token probability is evaluated at the corresponding response prefix. The mixture density associated with the equal allocation is

$$M(y \mid x_q) = \tfrac{1}{2} P_0(y \mid x_q) + \tfrac{1}{2} Q_\eta(y \mid x_q). \tag{10}$$

We evaluate both sequence densities for every optimization response, regardless of which proposal generated it. The raw importance ratios and self-normalized weights are

$$\omega_i = \frac{P_0(y_i \mid x_q)}{M(y_i \mid x_q)}, \qquad w_i = \frac{\omega_i}{\sum_{j=1}^{G} \omega_j}, \tag{11}$$

Here G is the group size, and densities include each proposal's sampling and stopping rules. Since $M \geq P_0/2$, $0 \leq \omega_i \leq 2$. Raw ratios weight the loss; w_i weights routing statistics. Unguided-only groups use $\omega_i = 1$, $w_i = 1/G$. Appendix B.4 specifies density evaluation and overlap checks; Appendix C gives the scope of correction.

4.3 CAPABILITY-RESIDUAL CREDIT ROUTING

Capability-Residual Credit Routing uses a **Capability Residual Projection Policy** to retain target-specific distinctions relative to a channel-balanced baseline. It partially attenuates their shared component, scales the residual by diagnosed need, and applies a group-level guard.

Constructing target-specific credit. We standardize valid criterion scores across the G responses with unweighted moments to obtain a_{ir}; invalid judgments and inactive criteria contribute zero. Fixed within-channel weights $p_q(r \mid c)$ sum to one without redistribution when criteria become inactive. Channel advantages and the baseline are

$$C_{ic} = \sum_{r \mapsto c} p_q(r \mid c) a_{ir}, \qquad B_i = \frac{1}{|\mathcal{C}_q|} \sum_{c \in \mathcal{C}_q} C_{ic}. \tag{12}$$

Appendix B.3 specifies criterion eligibility and standardization. For a diagnosed target, we compare its advantage with this baseline and form their residual difference:

$$T_i = C_{i,c_q^*}, \qquad R_i = T_i - B_i. \tag{13}$$

Projecting and scaling the residual. To control overlap with baseline credit, we use the self-normalized weights from Section 4.2 and define $\langle a, b \rangle_w = \sum_i w_i a_i b_i$ and $\|a\|_w = \sqrt{\langle a, a \rangle_w}$. We partially attenuate the baseline-aligned component and scale the resulting residual:

$$R_i' = R_i - \xi \frac{\langle R, B \rangle_w}{\langle B, B \rangle_w} B_i, \tag{14}$$

$$H_i = B_i + \lambda_q r_{\mathrm{ref}} \frac{R_i'}{\|R'\|_w}. \tag{15}$$

Here, $0 < \xi < 1$ controls the degree of partial projection. We set $R' = R$ when baseline energy is negligible. The coefficient $\lambda_q = \chi_q f(h_q^*)$ links correction strength to the diagnosed deficit and reliability. The schedule f is fixed, bounded, and nondecreasing, with $f(0) = 0$. The positive reference scale r_{ref} is computed from the weighted norms of the baseline and channel advantages. Appendix B.3 gives both definitions.

When applied, the correction has weighted norm $\lambda_q r_{\mathrm{ref}}$. We set $H = B$ when routing is disabled, the question has only one channel, or the weighted residual norm is below a numerical tolerance.

Selecting the final credit assignment. The guard checks the final normalized advantages used for policy optimization. Let $\mathcal{N}_w$ denote weighted advantage normalization with centering and magnitude control, as specified in Appendix B.3. We compute

$$A^0 = \mathcal{N}_w(B), \qquad A^1 = \mathcal{N}_w(H), \qquad \Delta A = A^1 - A^0. \tag{16}$$

We distinguish target opportunities from response preferences shared with the baseline:

$$\begin{aligned} \mathcal{U} &= \{(i,j) : T_i > T_j,\ B_i \le B_j\}, \\ \mathcal{P} &= \{(i,j) : B_i \ge B_j,\ T_i \ge T_j\}. \end{aligned} \tag{17}$$

We accept the candidate only when all four conditions hold:

$$\begin{aligned} &\mathcal{U} \neq \varnothing, \qquad \langle T, \Delta A \rangle_w > 0, \\ &A_i^1 \ge A_j^1 \quad \text{for every } (i,j) \in \mathcal{P}, \\ &\|\Delta A\|_w \le \frac{\|R'\|_w}{r_{\mathrm{ref}}}. \end{aligned} \tag{18}$$

The guard requires a target opportunity and a target-aligned change, preserves shared orderings, and bounds the adjustment. With acceptance $g_q \in \{0, 1\}$, select

$$A_i = g_q A_i^1 + (1 - g_q) A_i^0. \tag{19}$$

No normalization follows selection; questions without a target use A^0. Appendix C.2 proves these within-group credit properties.

Updating the policy. Each selected advantage weights the tokens of its response. We accumulate gradients over one batch pass and take one optimizer step. With proposal weights, judgments, and

advantages held fixed, the implemented loss has the same local gradient as the following surrogate at $\theta = \theta_{s-1}$:

$$\mathcal{L}_{\mathrm{sp}}(\theta) = \frac{1}{|\mathcal{B}|G} \sum_{q \in \mathcal{B}} \sum_{i=1}^{G} \omega_{qi} \left[-\frac{A_{qi}}{L_{qi}} \sum_{t=1}^{L_{qi}} \log \pi_\theta(y_{qit} \mid x_q, y_{qi,<t}) + \beta_{\mathrm{KL}} \bar{k}_{qi}(\theta) \right]. \tag{20}$$

Here $\mathcal{B}$ is the question batch, L_{qi} the response length, and ω_{qi} the raw proposal ratio. The token-averaged penalty $\bar{k}_{qi}$ compares the current and frozen starting policies with strength β_{KL}. Appendix B.4 gives the implemented loss.

5 Experiments

We evaluate scientific question answering and capability development. Table 1 summarizes benchmark performance; we then examine component contributions, capability changes, and guidance strength.

5.1 Experimental Setup

Benchmarks and evaluation. We use 487 ResearchQA questions (Yifei et al., 2025), the 100-question SQAv2 test split (Bragg et al., 2026), GPQA-Diamond's 198 questions (Rein et al., 2024), and 500 science questions from RubricHub excluded from training (Li et al., 2026). SQAv2 is closed-book and averages ingredient recall and answer precision, excluding citation metrics. Open-ended responses are graded by Gemma-4-31B-it, an open-weight judge evaluated in JudgeArena (Lushtaku et al., 2026). A fixed 128-question RubricHub subset supports capability tracking and development monitoring. Appendix D.1 specifies scoring and generation protocols.

Table 1: **Scientific question-answering scores.** Upper blocks provide external references; Qwen3-4B methods use our evaluation. Initial is the starting checkpoint. Subscripts are differences from Initial: score points, or percentage points for GPQA-Diamond. Within Qwen3-4B, bold/yellow-underlined scores mark best/second-best, including ties. Purple marks Ours; green/red subscripts mark gains/drops.

Method	ResearchQA ↑	SQAv2 ↑	GPQA-Diamond ↑	RubricHub ↑
Closed-source Frontier				
GPT-5	77.60	—	85.70	—
Claude Opus 4.1	74.30	86.10	77.30	—
GPT-4.1	70.80	—	50.50	—
DeepSeek V3.1	75.90	—	68.30	—
Open-weight				
Search-R1-7B	27.90	44.35	—	35.22
WebThinker-32B-DPO	48.60	65.80	—	46.34
WebExplorer-8B	64.80	85.00	43.81	47.38
Qwen3-8B	64.90	80.75	60.60	51.74
Qwen3-4B				
Initial	64.49	81.44	42.80	50.82
SFT	69.37$_{+4.88}$	83.24$_{+1.80}$	44.07$_{+1.27}$	50.56$_{-0.26}$
RL(GRPO)	64.32$_{-0.17}$	83.12$_{+1.68}$	43.80$_{+1.00}$	52.36$_{+1.54}$
Rubric-RL	69.37$_{+4.88}$	84.02$_{+2.58}$	45.20$_{+2.40}$	54.93$_{+4.11}$
RuscaRL	69.27$_{+4.78}$	81.24$_{-0.20}$	41.92$_{-0.88}$	53.08$_{+2.26}$
Ours	**71.92**$_{+7.43}$	**85.84**$_{+4.40}$	**47.60**$_{+4.80}$	**55.01**$_{+4.19}$

Baselines and training. Table 1 compares the starting Qwen3-4B checkpoint, SFT, RL, Rubric-RL, RuscaRL (Zhou et al., 2025), and RSRL. SFT uses 606 question–response pairs matched to the training questions; Rubric-RL averages standardized advantages over active criteria. RL comparisons share the starting checkpoint, question schedule, and optimization rollout budget. We train Qwen3-4B (Yang et al., 2025) with LoRA (Hu et al., 2022) on 2,460 RubricHub science questions for 100 updates, with 32 questions per update. Training criteria are scored by a local gpt-oss-120b judge, separately from evaluation. DeepSeek-V4-Pro constructs channels and mappings, which are frozen before training. Appendices A and D detail construction, configurations, and baseline implementations.

5.2 Overall Performance

Comparison with proprietary models. Against the external reference scores in Table 1, RSRL's 71.92 on ResearchQA exceeds GPT-4.1's 70.80. Its 85.84 on SQAv2 approaches Claude Opus 4.1's 86.10, highlighting the progress achieved from the 4B starting policy.

Comparison with open-weight models. RSRL's ResearchQA score exceeds the listed Qwen3-8B (64.90) and WebExplorer-8B (64.80) results. Its SQAv2 score also exceeds WebExplorer-8B's 85.00 and Qwen3-8B's 80.75, placing RSRL competitively among these larger reference systems on both open-ended benchmarks.

Comparison with post-training baselines. RSRL leads our Qwen3-4B comparison across all four benchmarks, ahead of SFT, RL, Rubric-RL, and RuscaRL. Gains over Rubric-RL are 2.55 points on ResearchQA, 1.82 on SQAv2, 2.40 percentage points on GPQA-Diamond, and 0.08 points on RubricHub.

5.3 Ablation Studies

Component contributions. We retain capability channels and remove guided exploration or residual routing in turn (Table 2; Figure 2). The exploration ablation uses only unguided optimization responses; the routing ablation retains guided sampling and proposal correction but uses the normalized channel-balanced baseline. Appendix D.4 specifies the comparison.

Targeted exploration. Removing guided exploration lowers Final Worst by 5.83 points and Overall by 7.03. Worst-Strong (Δ) is -0.32, compared with $+1.38$ for full RSRL, linking targeted exploration to stronger performance across weak and initially strong capabilities on this probe.

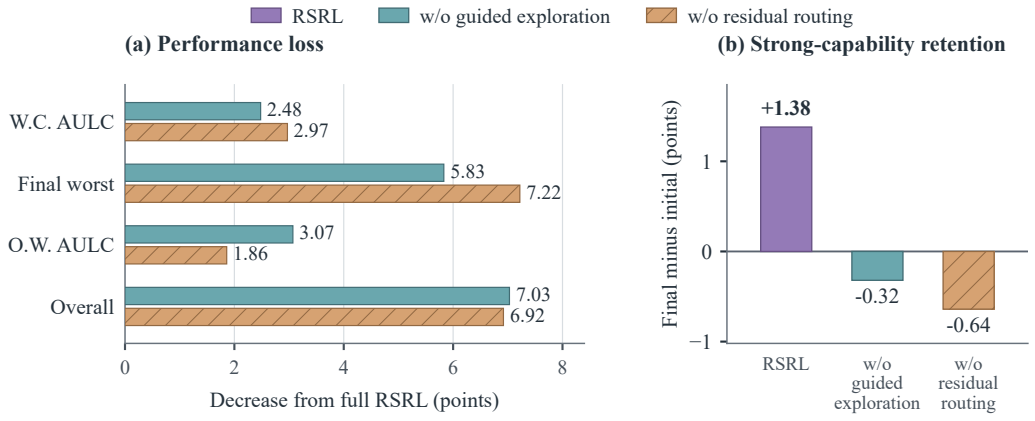

Figure 2: **Effects of component removal.** Values are derived from Table 2. (a) Decreases from full RSRL in the four performance metrics. (b) Worst-Strong (Δ), the minimum final-minus-initial change among initially strong capabilities; zero denotes no change.

Table 2: **Component ablations on the 128-question development probe.** Metrics follow Appendix D.1; scores/AULCs use 0–100 and changes use score points. Bold marks column maxima. Purple highlights RSRL; rose marks decreases from full RSRL.

Method	W.C.	F.W.	O.W.	W.S.	Overall
RSRL	**45.34**	**49.43**	**55.47**	**+1.38**	**55.40**
w/o guided exploration	42.86	43.60	52.40	−0.32	48.37
w/o residual routing	42.37	42.21	53.61	−0.64	48.48

Residual credit routing. Removing routing lowers Final Worst by 7.22 points and Overall by 6.92, with Worst-Strong (Δ) at -0.64. This highlights the contribution of capability-specific credit allocation.

5.4 Detailed Analysis

Table 3: **Capability tracking on 128 development questions.** W.C./O.W.: current-worst/initially weak channel AULC; F.W.: final worst score; W.S.: minimum strong-channel change; Overall: mean question score. Scores/AULCs use 0–100; changes use points. Bold marks maxima; purple highlights RSRL; green/red marks the W.S. sign.

Method	W.C.	F.W.	O.W.	W.S.	Overall
Rubric-RL	44.23	47.06	55.02	−0.45	54.31
RuscaRL	44.99	45.20	**56.08**	−2.31	54.00
RSRL	**45.34**	**49.43**	55.47	**+1.38**	**55.40**

Capability development. We track six channels at 27 checkpoints on the 128-question development probe (Figure 3). RSRL's Final Worst exceeds Rubric-RL and RuscaRL by 2.37 and 4.23 points, with Worst-Ch. AULC gains of 1.11 and 0.35 (Table 3). Its Worst-Strong (Δ) of $+1.38$ means every initially strong channel improves; Overall gains are 1.09 and 1.40. Appendix D.1 defines the metrics.

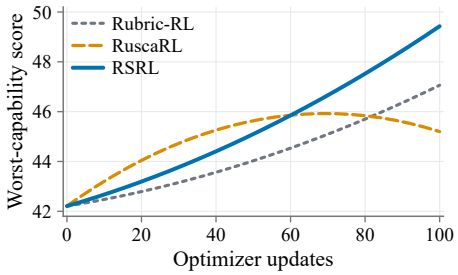

Figure 3: Worst-channel scores over 100 updates: 27 checkpoints on 128 development questions.

Guidance strength and proposal overlap. Offline diagnostic compares $1\times$, $3\times$, and $4\times$ guidance on 900 common-valid triggered questions.Mean target gains are 0.001747, -0.004924, and 0.012816 on the raw 0–1 scale, while median ESS/8 is 0.990510, 0.920261, and 0.865988. The paired $4\times - 1\times$ gain is 0.011069 (95% bootstrap interval $[-0.00402, 0.02641]$). These diagnostics characterize how guidance changes both target scores and proposal overlap, motivating their joint monitoring. Appendix E reports the full statistics and the separate 50-update comparison.

6 Conclusion

We introduced RSRL, which uses rubric structure to connect capability diagnosis, selective exploration, and response-level credit assignment. A shared capability target determines where exploration support is needed and how the resulting signals enter policy updates under the original question. Our scientific question-answering experiments show higher benchmark scores and improved weak-capability performance, alongside gains in initially strong capabilities on the development probe.

ETHICS STATEMENT

This work uses existing question and rubric datasets and does not introduce a human-subject study. Model-generated scientific answers can be incorrect, omit relevant evidence, or express unjustified confidence. Improvements on rubric-based evaluations should not be taken as validation for scientific or other high-stakes decisions. Automated channel construction and grading may reproduce biases in the underlying models and rubric design. Dataset access and any redistribution must respect the source licenses.

REPRODUCIBILITY STATEMENT

The manuscript and appendices describe the method, training settings, evaluation protocols, and aggregate results. The source package includes numerical tables and plotting code; the appendices provide proofs of the stated proposal and routing properties. Training code, frozen checkpoints, and per-response judgments are not included. Appendix F details domain aggregation and routing reliability.

AI USE STATEMENT

ChatGPT and Codex were used for literature search, discussion of conceptual framing and mathematical interpretation. The experimental pipeline also uses an LLM for offline channel construction and LLM judges for training and evaluation, as described in the paper. The human authors remain responsible for the final claims, citations, experimental provenance, and submitted content

REFERENCES

Rahul K. Arora, Jason Wei, Rebecca Soskin Hicks, Preston Bowman, Joaquin Quiñonero-Candela, Foivos Tsimpourlas, Michael Sharman, Meghan Shah, Andrea Vallone, Alex Beutel, Johannes Heidecke, and Karan Singhal. HealthBench: Evaluating large language models towards improved human health. *arXiv preprint arXiv:2505.08775*, 2025. URL https://arxiv.org/abs/2505.08775.

Jonathan Bragg, Mike D'Arcy, Nishant Balepur, Dan Bareket, Bhavana Dalvi, Sergey Feldman, Dany Haddad, Jena D. Hwang, Peter Jansen, Varsha Kishore, et al. AstaBench: Rigorous benchmarking of AI agents with a scientific research suite. In *International Conference on Learning Representations*, 2026. URL https://arxiv.org/abs/2510.21652.

Alex Graves, Marc G. Bellemare, Jacob Menick, Rémi Munos, and Koray Kavukcuoglu. Automated curriculum learning for neural networks. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70, pp. 1311–1320. PMLR, 2017. URL https://proceedings.mlr.press/v70/graves17a.html.

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

Vipul Gupta, Zihao Wang, Razvan-Gabriel Dumitru, MohammadHossein Rezaei, Aakash Sabharwal, and Yunzhong He. CRAFT: Clustering rubrics to diagnose weak LLM capabilities and generate targeted fine-tuning data. *arXiv preprint arXiv:2607.16122*, 2026. URL https://arxiv.org/abs/2607.16122.

Jinkun Hou, Zhuo Liu, Huimin Ren, Hongsheng Xin, Pan Zhou, and Kun Zhan. RISE-RL: Rubric-informed selective exploration for open-ended reinforcement learning. *arXiv preprint arXiv:2608.09123*, 2026. URL https://arxiv.org/abs/2608.09123.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*, 2022. URL https://arxiv.org/abs/2106.09685.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. Prometheus: Inducing fine-grained evaluation capability in language models. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=8euJaTveKw.

Sunzhu Li, Jiale Zhao, Miteto Wei, Huimin Ren, Yang Zhou, Jingwen Yang, Shunyu Liu, Kaike Zhang, and Wei Chen. RubricHub: A comprehensive and highly discriminative rubric dataset via automated coarse-to-fine generation. *arXiv preprint arXiv:2601.08430*, 2026. URL https://arxiv.org/abs/2601.08430.

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12286–12312. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.acl-long.687. URL https://aclanthology.org/2023.acl-long.687/.

Haotian Liu, Yihao Liu, Jingwei Ni, Siyuan Huang, Xinpeng Liu, Pengyu Cheng, Jiajun Song, Ruijin Ding, Junfeng Li, Zhechao Yu, Mengyu Zhou, Hongteng Xu, Xiaoxi Jiang, and Guanjun Jiang. GD^{2}PO: Mitigating multi-reward conflicts via group-dynamic reward-decoupled policy optimization. *arXiv preprint arXiv:2606.16771*, 2026a. URL https://arxiv.org/abs/2606.16771.

Shih-Yang Liu, Xin Dong, Ximing Lu, Shizhe Diao, Peter Belcak, Mingjie Liu, Min-Hung Chen, Hongxu Yin, Yu-Chiang Frank Wang, Kwang-Ting Cheng, Yejin Choi, Jan Kautz, and Pavlo Molchanov. GDPO: Group reward-decoupled normalization policy optimization for multi-reward RL optimization. *arXiv preprint arXiv:2601.05242*, 2026b. URL https://arxiv.org/abs/2601.05242.

Erlis Lushtaku, Bora Kargi, Ali Elganzory, Fabio Ferreira, Alejandro R. Salamanca, Julia Kreutzer, and David Salinas. JudgeArena: A unified framework for reproducible LLM-judge evaluation. *arXiv preprint arXiv:2608.02620*, 2026. URL https://arxiv.org/abs/2608.02620.

Jonathan Pei, Kevin Yang, and Dan Klein. PREADD: Prefix-adaptive decoding for controlled text generation. In *Findings of the Association for Computational Linguistics: ACL 2023*, pp. 10018–10037. Association for Computational Linguistics, 2023. doi: 10.18653/v1/2023.findings-acl.636. URL https://aclanthology.org/2023.findings-acl.636/.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof Q&A benchmark. In *First Conference on Language Modeling*, 2024. URL https://arxiv.org/abs/2311.12022.

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 *International Conference on Machine Learning*, 2026. URL https://arxiv.org/abs/2511.19399.

Eric Veach and Leonidas J. Guibas. Optimally combining sampling techniques for Monte Carlo rendering. In *Proceedings of SIGGRAPH*, pp. 419–428, 1995. URL https://graphics.stanford.edu/papers/combine/.

Wen Wang, Jiahua Bao, Tu Yongsiqi, Yihao Liu, Haotian Zhou, Haoxuan Ma, Mengyu Zhou, Wenkui Fan, Junwei He, Xiaoxi Jiang, and Guanjun Jiang. SMOPD: Multi-reward reinforcement learning via specialize-and-merge online policy distillation. *arXiv preprint arXiv:2608.03092*, 2026. URL https://arxiv.org/abs/2608.03092v2.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025. URL https://arxiv.org/abs/2505.09388.

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

Yang Zhou, Sunzhu Li, Shunyu Liu, Wenkai Fang, Kongcheng Zhang, Jiale Zhao, Jingwen Yang, Yihe Zhou, Jianwei Lv, Tongya Zheng, Hengtong Lu, Wei Chen, Yan Xie, and Mingli Song. Breaking the exploration bottleneck: Rubric-scaffolded reinforcement learning for general LLM reasoning. *arXiv preprint arXiv:2508.16949*, 2025. URL https://arxiv.org/abs/2508.16949v7.

A CAPABILITY CONSTRUCTION AND DIAGNOSTIC SCORES

A.1 FROZEN DOMAIN AND LOCAL CHANNELS

We construct the domain codebook from 64 questions sampled from training, calibration, and development records, excluding evaluation and reserve splits. Selection uses a fixed hash ordering. The construction model receives the questions and their source rubrics. It proposes three to eight recurring capabilities, after which coverage checks and consolidation yield six science channels: quantitative and formal derivation; precise technical communication and structured setup; mechanistic and causal explanation; experimental, computational, and intervention design; comparison and synthesis across alternatives; and assessment of evidence, assumptions, and scope.

Each question's source criteria are grouped into two to five local channels, with at least two criteria per channel. Canonical source identifiers prevent aliases from receiving duplicate assignments. Programmatic checks enforce coverage and unique assignment; model-based review checks grouping coherence. The frozen training set contains 2,460 questions after construction checks. These local channels retain question-specific requirements while the domain codebook supplies shared categories.

A.2 MAPPING LOCAL CHANNELS TO THE DOMAIN CODEBOOK

Construction and mapping use the API identifier `deepseek-v4-pro`, with temperature zero, and JSON output. The mapping judge receives the question, local channel definition, and domain codebook. It returns a domain identifier, applicability decision, confidence score, and rationale.

For each local channel, two separate requests to the same model produce mappings $m_q^A(c), m_q^B(c)$ and confidence scores $p_{q,c}^A, p_{q,c}^B$. We accept a mapping when both requests judge it applicable and

$$m_q^A(c) = m_q^B(c) \neq \perp, \qquad \min(p_{q,c}^A, p_{q,c}^B) \geq 0.8.$$

The accepted confidence is $m_{q,c}^{\text{conf}} = \min(p_{q,c}^A, p_{q,c}^B)$. Otherwise, the channel remains local-only, with $m_q(c) = \perp$ and $m_{q,c}^{\text{conf}} = 0$. These are two calls to one model, and their self-reported scores are not calibrated probabilities. Definitions, assignments, and mapping confidences are frozen before policy training. Appendix F describes their use in diagnosis.

A.3 RUBRIC BANKS AND VALID OBSERVATIONS

Within each channel, diagnostic and training banks use the same set of at most five canonical criteria, selected by descending absolute source points with fixed hash tie-breaking. The evaluation bank retains the source criteria required by its evaluation protocol. Evaluation criteria are not supplied as generation guidance.

Let $s_{jr} \in [0, 1]$ be the polarity-aligned diagnostic score for response j and criterion r, and let v_{jr} indicate validity. Scores are normalized by the criterion's predefined range. A valid zero remains an observation. For $N_r^D = \sum_j v_{jr} > 0$, define

$$\mu_r^D = \frac{\sum_j v_{jr} s_{jr}}{N_r^D}, \qquad \mathcal{F}_{q,c} = \{r \in \mathcal{R}_{q,c}^D : N_r^D > 0\}.$$

Using frozen criterion reliability u_r, the local score and reliability are

$$S_{q,c} = \frac{\sum_{r \in \mathcal{F}_{q,c}} u_r \mu_r^D}{\sum_{r \in \mathcal{F}_{q,c}} u_r}, \qquad r_{q,c} = \frac{1}{|\mathcal{F}_{q,c}|} \sum_{r \in \mathcal{F}_{q,c}} u_r.$$

Thus, a wholly missing criterion is omitted from both numerator and denominator. The science banks use $u_r = 1$, giving equal weight to available criterion means. If $\mathcal{F}_{q,c}$ is empty, the local score is unavailable and $r_{q,c} = 0$; diagnosis then follows the missing-evidence rules in Section 4.1.

The count used for historical pooling is

$$n_{q,c} = \sum_{j=1}^{4} \mathbf{1}\big[\exists r \in \mathcal{R}_{q,c}^D : v_{jr} = 1\big].$$

One valid criterion suffices to count a response once. This count differs from the stricter criterion eligibility tests for guided exploration and optimization advantages below.

B TRAINING PROCEDURE AND ESTIMATOR DETAILS

B.1 ONE POLICY UPDATE

An unguided diagnostic pass initializes the domain state without gradient updates. Each subsequent update uses the following order:

1. Snapshot the policy and committed domain state. Generate four unguided diagnostic responses for each scheduled question.
2. Score the diagnostic criteria, pool local and historical evidence, and select the target capability. Hold this target fixed for the question's update.
3. Apply the exploration support test. Generate four unguided and four guided optimization responses if active, or eight unguided responses otherwise. Evaluate the proposal densities and weights.
4. Score the optimization responses, construct the baseline and routed candidate, and apply the guard after final normalization.
5. Accumulate the clean-context loss over the question batch, clip the accumulated gradient, and take one optimizer step.
6. Aggregate the unguided diagnostic observations within questions and then equally across questions. Commit the domain update for use at the next step.

All questions in an update use the same historical snapshot. Guided responses do not update the domain state.

B.2 SUPPORT DETECTION AND EXPLORATION STRENGTH

For the selected channel, let $\mathcal{E} = \{r : N_r^D \geq 4\}$. Since diagnosis uses four responses, only criteria with all four valid judgments enter the support test. For each eligible criterion, compute its population variance V_r and positive-response fraction p_r, using score 0.5 as the positive threshold. The ZERO regime requires all eligible scores to be zero. The LOW regime requires at least one $V_r > 10^{-4}$ and $|\mathcal{E}|^{-1} \sum_{r \in \mathcal{E}} p_r \leq 0.25$. An empty eligible set or any other support state leaves sampling unguided.

The coefficient in Section 4.2 is

$$\eta_q = \frac{\eta_{\max}}{\sqrt{H_{\max}}} \operatorname{clip}\left(\frac{h_q^* - 0.05}{0.30 - 0.05}, 0, 1 \right) g_{\text{support}}, \tag{21}$$

where g_{support} is one for ZERO, one-half for LOW, and zero otherwise. Here $H_{\max} = 3072$ is the response cap and $\eta_{\max} = 1$, so the effective coefficient is capped at $1/\sqrt{3072}$. The strength therefore vanishes for deficits at most 0.05 and saturates at 0.30.

The target prompt contains the selected capability's description and associated training criteria, together with an instruction not to disclose the internal guidance. The control prompt supplies nondirectional guidance. The three distributions are evaluated at the same response prefix; the optimization loss is always conditioned on the original question.

B.3 CRITERION ADVANTAGES AND FINAL NORMALIZATION

Let z_{ir} be an optimization score oriented so that higher is better, with validity v_{ir}. For a criterion with $N_r = \sum_i v_{ir} > 0$, compute unweighted moments

$$\mu_r = \frac{\sum_i v_{ir} z_{ir}}{N_r}, \qquad \sigma_r^2 = \frac{\sum_i v_{ir} (z_{ir} - \mu_r)^2}{N_r}. \tag{22}$$

A criterion is active if it is canonical, has frozen reliability at least 0.8, has at least $\max(4, \lceil 0.75G \rceil) = 6$ valid observations, and has standard deviation greater than 0.05 times its

predefined score range. Valid active entries use $a_{ir} = \mathrm{clip}((z_{ir} - \mu_r)/\sigma_r, -5, 5)$; all other entries are zero. Fixed within-channel weights are not redistributed when criteria become inactive.

The deficit schedule and reference scale in Section 4.3 are

$$f(h) = \lambda_{\max}\left[\mathrm{clip}\left(\frac{h - h_R}{1 - h_R}, 0, 1\right)\right]^p, \tag{23}$$

$$\|C\|_w^2 = \sum_i w_i \frac{1}{|\mathcal{C}_q|}\sum_c C_{ic}^2, \qquad r_{\mathrm{ref}} = \max\big(\|B\|_w, \min(1, \|C\|_w), \varepsilon\big). \tag{24}$$

We set $R' = R$ when $\langle B, B\rangle_w \le \varepsilon^2$ and omit the correction when $\|R'\|_w \le 10^{-6}$. Single-channel questions and zero routing strength use the baseline.

The final normalization is

$$\widehat{X}_i = \mathrm{clip}\left(\frac{X_i}{\max(\|X\|_w, \varepsilon)}, -5, 5\right), \qquad V_i = \widehat{X}_i - \sum_j w_j \widehat{X}_j, \tag{25}$$

$$Z_i = \frac{V_i}{\max(\|V\|_w, \varepsilon)}, \qquad \mathcal{N}_w(X)_i = Z_i \min\left(1, \frac{5}{\max_j |Z_j|}\right). \tag{26}$$

The last factor is one when $Z = 0$. Both candidate and baseline use this order of operations. No further normalization follows the guard decision. Table 5 lists the fixed parameters.

B.4 Sequence densities and the implemented loss

Training uses temperature one, top-p one, and no top-k truncation. The target–control contrast uses full softmax probabilities. Sequence log densities include all realized tokens, including sampled EOS, but exclude padding and any unobserved EOS at the length cap. Teacher-forced replay evaluates the guided density of unguided responses, and log-sum-exp evaluates the mixture density.

We monitor $\mathrm{ESS} = (\sum_i \omega_i)^2 / \sum_i \omega_i^2$ and guided mass $\mathrm{Share}_g = \sum_{i\ \mathrm{guided}} w_i$. Valid branch densities and $\mathrm{ESS}/G \ge 0.25$ are required; failure raises an error without modifying the weights. Small guided weights are not floored.

Write $\ell_{qit}(\theta) = \log \pi_\theta(y_{qit} \mid x_q, y_{qi,<t})$. With all sampled quantities held fixed, the implemented loss is

$$\mathcal{L}_{\mathrm{impl}}(\theta) = \frac{1}{|\mathcal{B}|G}\sum_{q,i}\omega_{qi}\left[-\frac{1}{L_{qi}}\sum_t \min(r_{qit}A_{qi}, \mathrm{clip}(r_{qit}, 0.8, 1.2)A_{qi}) + \beta_{\mathrm{KL}}\bar{k}_{qi}(\theta)\right], \tag{27}$$

$$r_{qit} = \exp\big(\ell_{qit}(\theta) - \mathrm{stopgrad}(\ell_{qit}(\theta))\big), \tag{28}$$

$$\bar{k}_{qi}(\theta) = \frac{1}{L_{qi}}\sum_t \left(e^{d_{qit}(\theta)} - d_{qit}(\theta) - 1\right),$$

$$d_{qit}(\theta) = \log \pi_{\mathrm{ref}}(y_{qit} \mid x_q, y_{qi,<t}) - \ell_{qit}(\theta). \tag{29}$$

Here stopgrad denotes stop-gradient and π_{ref} is the frozen starting policy. During the single pre-update pass, $r_{qit} = 1$ with nonzero derivative, so clipping is inactive and the policy term has the local gradient stated in Equation (20). The reference penalty remains differentiable. Raw proposal weights multiply both terms. This equality describes the specified one-pass update.

C Properties of exploration and routing

C.1 Proposal support and importance correction

At a fixed prefix, the bound $|\delta(v)| \le b$ places both $\exp(\eta\delta(v))$ and its normalizer in $[e^{-\eta b}, e^{\eta b}]$. Hence

$$e^{-2\eta b} \le \frac{q_\eta(v)}{p_0(v)} \le e^{2\eta b} \quad \text{for } p_0(v) > 0. \tag{30}$$

Finite perturbations preserve probability support. The diagnostic ZERO label concerns observed scores, not zero probability under the policy.

Conditioning on diagnosis fixes P_0 and Q_η. For an integrable function f independent of the sampled optimization group, the equal-allocation balance estimator satisfies

$$\frac{1}{2}\mathbb{E}_{P_0}\left[\frac{P_0}{M}f\right] + \frac{1}{2}\mathbb{E}_{Q_\eta}\left[\frac{P_0}{M}f\right] = \int M(y)\frac{P_0(y)}{M(y)}f(y)\,dy = \mathbb{E}_{P_0}[f]. \tag{31}$$

Also, $M \geq P_0/2$ gives the ratio bound in Section 4.2. This identity applies to a fixed integrand; criterion normalization and routing depend on the sampled group, so it does not establish unbiasedness of the complete gradient estimator.

For fixed p_0 and δ, let $Z(\eta) = \mathbb{E}_{p_0} e^{\eta\delta}$. Then

$$D_{\mathrm{KL}}(p_0\|q_\eta) = \log Z(\eta) - \eta\mathbb{E}_{p_0}\delta, \qquad \frac{d^2}{d\eta^2}D_{\mathrm{KL}}(p_0\|q_\eta) = \mathrm{Var}_{q_\eta}(\delta) \geq 0. \tag{32}$$

The first derivative is zero at $\eta = 0$, so this divergence is nondecreasing for $\eta \geq 0$ at the fixed prefix. This is the setting used for the replay diagnostic in Appendix E.

C.2 NORMALIZATION AND GUARD GUARANTEES

Assume finite channel advantages, $w_i \geq 0$, $\sum_i w_i = 1$, and the numerical conventions in Appendix B.3. Partial projection satisfies $\langle R', B\rangle_w = (1-\xi)\langle R, B\rangle_w$ on its nondegenerate branch; it attenuates the baseline-parallel component. When applied, the correction has $\|H - B\|_w = \lambda_q r_{\mathrm{ref}}$.

Proposition C.1. *For every finite X, $\mathcal{N}_w(X)$ has weighted mean zero, weighted norm at most one, and coordinate magnitude at most five. It preserves weak coordinate ordering.*

Proof. The first positive common scaling and nondecreasing clipping preserve order. Subtracting a common weighted mean centers the vector without changing pairwise differences. Division by $\max(\|V\|_w, \varepsilon)$ gives weighted norm at most one. The final common factor is positive and at most one, preserves centering and order, and enforces the coordinate bound. □

Proposition C.2. *The advantage A selected in Equation* (19) *has the preceding normalization properties. For every $(i,j) \in \mathcal{P}$, $A_i \geq A_j$. In addition,*

$$\langle T, A - A^0\rangle_w \geq 0, \qquad r_{\mathrm{ref}}\|A - A^0\|_w \leq \|R'\|_w.$$

The first inequality is strict whenever the candidate is accepted.

Proof. On acceptance, the ordering and both inequalities are explicit guard conditions. On rejection, $A = A^0$: its ordering follows that of B, and its change is zero. Either selected vector has the normalization properties, and no subsequent transformation is applied. □

A constant target has no strict target preference. The same opportunity set is empty for $T = aB + b\mathbf{1}$ with $a > 0$. Both cases return A^0. Moreover, if $\|R'\|_w / r_{\mathrm{ref}} \to 0$, the weighted norm of the final credit change tends to zero.

C.3 BOUNDARY AND ACCEPTANCE EXAMPLES

Let eight responses have equal weights and define orthogonal vectors with unit weighted norm

$$u = (-1,-1,-1,-1,1,1,1,1), \qquad v = (-1,-1,1,1,-1,-1,1,1).$$

For $C_1 = 0$, $C_2 = u$, and $C_3 = -0.8u + 0.2v$, selecting channel one gives $T = 0$, $B = (0.2u + 0.2v)/3$, $\|B\|_w = 0.094281$, and $r_{\mathrm{ref}} = 0.748331$. With $\xi = 0.5$ and $\lambda_q = 0.45$, the unguarded candidate is $H = -2.571764B$. The guard returns A^0 because there is no target opportunity. More generally, for $T = aB$ with $0 \leq a < 1$ and an applied correction,

$$H = \left(1 - \frac{\lambda_q r_{\mathrm{ref}}}{\|B\|_w}\right) B.$$

This exposes the aligned-target boundary even when the target varies across responses.

For an accepted case, take $C_1 = u$, $C_2 = v$, and $T = u$. Then $B = (u+v)/2$, $R' = (u-v)/2$, and $r_{\text{ref}} = 1$. At $\lambda_q = 0.45$,

$$A^0 = (u+v)/\sqrt{2}, \qquad A^1 = 0.9761921u + 0.2169079v.$$

There is a target preference within a baseline tie, shared orderings are preserved, and $\langle T, \Delta A \rangle_w = 0.2690853 > 0$. The weighted norm of the change is $0.5591975 < 1/\sqrt{2}$, so all four conditions hold. These channel vectors can be formed from distinct binary criteria with repeated score patterns; the examples describe the rule's behavior, not experimental frequencies.

D Experimental settings and evaluation

D.1 Evaluation protocols and capability metrics

Evaluation responses use the original question without rubric or channel guidance. Open-ended scoring uses `Gemma-4-31B-it` with thinking enabled, a default context of 16,384 tokens, and an 8,192-token completion ceiling, separately from the training judge.

ResearchQA. The 487 evaluation questions, containing 3,633 source criteria, are disjoint from the 2,460 training questions. Source-weighted criterion scores are aggregated within each question and averaged equally over questions. Missing or invalid judgments receive zero. Generation uses temperature 0.6, top-p 0.95, top-k 20, an initial response cap of 16,384, and context length 40,960. The evaluation protocol includes repetition and length repair.

SQAv2. We use the 100-question SQA-CS-V2 test split with 1,094 source ingredients and ASTA/astabench 0.3.1 scoring functions. The reported content score is $100(\text{IR} + \text{AP})/2$, where ingredient recall and answer precision are on a 0–1 scale. This closed-book evaluation excludes citation metrics.

GPQA-Diamond. Accuracy is averaged across four independent evaluation runs on 198 questions. Generation uses temperature 0.7, top-p 0.8, top-k 20, a 32,768-token response cap, and context length 40,960. Format completion handles answers without a parseable option without consulting the answer key; remaining unparseable answers count as incorrect. These are inference repetitions of each trained model.

RubricHub and capability tracking. The 500-question science evaluation set is excluded from policy training. Its evaluator averages valid criterion judgments within a question, omitting invalid judgments, and then averages across questions. The fixed 128-question development probe is a subset of this set. It is evaluated at checkpoints $0, 2, 4, 8, 12, \ldots, 100$, giving 27 observations.

Let $S_z(s)$ average the scores of questions applicable to capability z at checkpoint s, and $W(s) = \min_z S_z(s)$. For $0 = s_0 < \cdots < s_m = S$, normalized AULC is

$$\text{AULC}_S(W) = \frac{1}{S} \sum_{j=1}^{m} (s_j - s_{j-1}) \frac{W(s_j) + W(s_{j-1})}{2}. \tag{33}$$

The main comparison uses $S = 100$. Orig.-Weak AULC replaces W with the score of the fixed initially weak capability, assessment of evidence, assumptions, and scope. The prespecified strong set comprises technical communication, mechanistic explanation, and quantitative derivation. These identities come from the shared initial training diagnosis. The retention metric is

$$\Delta_{\text{strong}} = \min_{z \in \mathcal{Z}_{\text{strong}}} \big[S_z(100) - S_z(0) \big].$$

Overall is the mean question-level response score, not a mean over the six capabilities.

D.2 Training configuration

Each RL run processes 3,200 question visits. RSRL generates 25,600 optimization responses and 12,800 diagnostics, excluding initialization. The 256 optimization trajectories per update are processed in microbatches of four before one optimizer step. Runs use one to three A800 GPUs across

Table 4: Training configuration. Response budgets are per question. The lower block records additional RuscaRL settings.

Setting	Value
Starting policy / training questions	Qwen3-4B / 2,460 science questions
Optimizer updates / questions per update	100 / 32
Diagnostic / optimization responses	4 / 8
Active guided-sampling allocation	4 unguided + 4 guided
Response cap / generation mode	3,072 tokens / non-thinking
LoRA rank / alpha / dropout	32 / 32 / 0
LoRA targets / bias	All linear layers / none
Learning rate / KL coefficient	10^{-5} / 10^{-3}
Trajectory microbatch / gradient clip	4 / 1.0
Training judge	`gpt-oss-120b`, medium reasoning
Judge completion ceiling	8,192 tokens
Additional RuscaRL settings	
Scaffold decay / intra-group allocation	Sigmoid ($\alpha = 125$, $t_0 = 0.2$) / linear
Rollout / training contexts	Scaffolded / original question
Criterion score levels	0, 0.5, 1
Sampling	Temperature 1; top-p 1; no top-k
Optimizer / policy clip	AdamW / 0.2
Checkpoint interval / capability guidance and routing	Every 2 updates / neither used

stages, with judging on separate resources. Equal optimization rollout budgets do not equate the additional computation for the guided proposal's conditional distributions and density evaluation.

Table 5: Fixed diagnosis, exploration, and routing parameters.

Parameter	Value
Capability threshold τ / pooling κ	0.75 / 8
Domain retention β_D / mapping acceptance	0.9 / 0.8
Guidance activation / saturation deficit	0.05 / 0.30
Guidance $\eta_{\max}$ / contrast clip b	1 / 4
Positive-score threshold / Low fraction	0.5 / 0.25
Partial projection ξ	0.5
Routing $\lambda_{\max}$ / h_R / exponent p	0.45 / 0.05 / 1.5
Criterion / final advantage cap	5 / 5
Normalization ε / residual tolerance	10^{-8} / 10^{-6}

All settings remain fixed during training. With $\kappa = 8$, four valid local diagnostics receive one-third of the pooled weight and an available domain prior receives two-thirds.

D.3 Baseline details

SFT uses 606 normalized exact-query matches to the training questions, selecting the highest-scoring existing response from a best-of-six source. It trains for one epoch with effective batch size eight, microbatch two, four accumulation steps, learning rate 10^{-5}, and LoRA rank/alpha 32/32. Loss is restricted to assistant tokens.

Rubric-RL standardizes eligible criteria separately, averages active criterion advantages, and applies root-mean-square scaling and clipping to $[-5, 5]$. RSRL instead forms fixed-weight channel advantages before the balanced baseline.

RuscaRL uses linear intra-group scaffolding and sigmoid withdrawal over training progress. Criterion scores are first aggregated into a scalar reward for each response, then standardized across the eight optimization responses. Scaffolds appear during generation; training log probabilities use

the original question. Its model, LoRA, judge, learning rate, batch, length cap, and update schedule are given in Table 4. It generates four discovery responses per question and uses neither capability-guided exploration nor residual credit routing.

D.4 Component ablation protocol

The comparisons in Section 5.3 share the frozen capability channels, mappings, and diagnostic procedure. Removing guided exploration sets every optimization group to eight unguided responses, with unit proposal ratios, while retaining residual routing and its guard. Removing residual routing retains guided/unguided sampling and density evaluation but uses A^0 for every policy update. Thus the reported contrasts concern the guided proposal and routed credit rule under a fixed capability structure. The shared mapping and adaptive gate are not separately isolated by these variants.

E Exploration-strength diagnostics

This appendix provides the protocol and complete statistics for the guidance-strength analysis in Section 5.4. Offline diagnostic compares $1\times$, $3\times$, and $4\times$ guidance strength on 1000 triggered questions: 310 with Zero and 690 with Low support. We use the 900 questions valid under all three settings for paired summaries; Scores in Table 6 use the raw 0–1 scale and measure guided-minus-clean target or non-target performance. The effective-signal proxy is a signed sum of guided importance weights, response advantage, and target-score improvement; it is not a gradient norm.

Table 6: Offline guidance diagnostics on 900 common-valid questions. Gains use the 0–1 score scale; ESS is divided by the eight-response group size.

Statistic	$1\times$	$3\times$	$4\times$
Mean target gain	0.001747	-0.004924	0.012816
Mean non-target gain	-0.011115	-0.006001	0.002867
Median ESS / 8	0.990510	0.920261	0.865988
Mean guided weight share	0.489837	0.427703	0.381443
Mean effective-signal proxy	0.044115	0.039832	0.038713

Table 6 shows larger target-gain point estimates at $4\times$ together with lower proposal overlap. The paired $4\times - 1\times$ gain is 0.011069 with a recorded 95% bootstrap interval of $[-0.00402, 0.02641]$, which includes zero. At $4\times$, fixed-prefix sequence KL and ESS have Spearman correlation -0.8029. The separate 50-update comparison has a lower final worst-channel score but higher capability mean at $4\times$ (Table 7).

Target and non-target gains compare guided and unguided channel scores. The signed proxy is

$$U_q = \sum_{i \text{ guided}} w_i A_i \left(S_{i,c_q^*} - \overline{S}_{\text{clean},c_q^*} \right).$$

Here A_i is the advantage used in that diagnostic. Since two negative factors can contribute positively, this statistic is not a beneficial-gradient measure. The paired $4\times - 1\times$ target-gain comparison has 450 wins, 250 ties, and 200 losses, with median zero. Fixed-prefix replay gives mean sequence KL values of 0.03658, 0.33031, and 0.58828 at the three strengths.

The 50-update run starts from a fresh base checkpoint and uses the first 50 steps of the training schedule. Table 7 uses checkpoints. These settings make it an exploratory comparison rather than an isolated strength ablation.

Table 7: Earlier 50-update training comparison. AULCs are normalized by 50. Capability macro averages the six channel scores.

Method	Worst AULC	Final worst	Initial weak AULC	Strong $\Delta_{\min}$	Capability macro
Full $1\times$	42.6268	42.2695	52.6774	$+0.0620$	50.2890
Full $4\times$	42.3726	40.9658	52.9860	-1.2417	50.8242
RuscaRL	43.2669	44.9219	53.1383	-0.8908	51.1511

F Domain aggregation and routing reliability

Let $\mathcal{C}_{q,z}^{\text{valid}}$ contain the applicable local channels mapped to z with an available local score and $n_{q,c} > 0$. The score and confidence contribution of one question are

$$O_{q,z} = \frac{\sum_{c \in \mathcal{C}_{q,z}^{\text{valid}}} r_{q,c} S_{q,c}}{\sum_{c \in \mathcal{C}_{q,z}^{\text{valid}}} r_{q,c}}, \qquad D_{q,z} = \frac{\sum_{c \in \mathcal{C}_{q,z}^{\text{valid}}} r_{q,c} m_{q,c}^{\text{conf}} \min(1, r_{q,c})}{\sum_{c \in \mathcal{C}_{q,z}^{\text{valid}}} r_{q,c}}.$$

With unit local reliability these become the mean channel score and mean mapping confidence. Mapping confidence does not weight the score average.

Let $\mathcal{Q}_z^0$ be the initialization questions with an available contribution to z. Shared unguided diagnostics initialize

$$U_{z,0} = \frac{1}{|\mathcal{Q}_z^0|} \sum_{q \in \mathcal{Q}_z^0} O_{q,z}, \qquad d_z^{\text{conf}} = \frac{1}{|\mathcal{Q}_z^0|} \sum_{q \in \mathcal{Q}_z^0} D_{q,z}.$$

During training, $\bar{U}_{z,s}$ averages the current $O_{q,z}$ equally over observed questions and updates the domain score through Equation (5). It uses local diagnostic observations rather than prior-pooled scores. Unobserved capabilities retain their previous estimates. Domain and mapping confidence remain fixed.

With both local evidence and a mapped domain estimate, routing uses

$$\chi_{q,c} = \text{clip}\left[\nu_{q,c} \min(1, r_{q,c}) + (1 - \nu_{q,c}) d_{m_q(c)}^{\text{conf}} m_{q,c}^{\text{conf}}, 0, 1\right], \qquad \nu_{q,c} = \frac{n_{q,c}}{n_{q,c} + \kappa}. \tag{34}$$

With local evidence alone, $\chi_{q,c} = \text{clip}(r_{q,c}, 0, 1)$; with a mapped domain estimate alone, $\chi_{q,c} = d_{m_q(c)}^{\text{conf}} m_{q,c}^{\text{conf}}$. Channels with neither source are excluded from target selection. The selected channel supplies χ_q. A prior alone does not satisfy the local support test for guided exploration. This composite weight controls routing strength; unit local reliability is a configured weight, not perfect judge accuracy.