

BREAKING THE EXPLORATION BOTTLENECK: RUBRIC-SCAFFOLDED REINFORCEMENT LEARNING FOR GENERAL LLM REASONING

Anonymous authors

Paper under double-blind review

ABSTRACT

Recent advances in Large Language Models (LLMs) have underscored the potential of Reinforcement Learning (RL) to facilitate the emergence of reasoning capabilities. Despite the encouraging results, a fundamental dilemma persists as RL improvement relies on learning from high-quality samples, yet the exploration for such samples remains bounded by the inherent limitations of LLMs. This, in effect, creates an undesirable cycle in which *what cannot be explored cannot be learned*. In this work, we propose Rubric-Scaffolding Reinforcement Learning (RuscaRL), a novel instructional scaffolding framework designed to break the exploration bottleneck for general LLM reasoning. Specifically, RuscaRL introduces checklist-style rubrics as (1) *explicit scaffolding* for exploration during rollout generation, where different rubrics are provided as external guidance within task instructions to steer diverse high-quality responses. This guidance is gradually decayed over time, encouraging the model to internalize the underlying reasoning patterns; (2) *verifiable rewards* for exploitation during model training, where we can obtain robust LLM-as-a-Judge scores using rubrics as references, enabling effective RL on general reasoning tasks. Extensive experiments demonstrate the superiority of the proposed RuscaRL across various benchmarks, effectively expanding reasoning boundaries under the Best-of-N evaluation. Notably, RuscaRL significantly boosts Qwen2.5-7B-Instruct from 23.6 to 50.3 on HealthBench-500, surpassing GPT-4.1. Furthermore, our fine-tuned variant on Qwen3-30B-A3B-Instruct achieves 61.1 on HealthBench-500, outperforming leading LLMs including OpenAI-o3.

1 INTRODUCTION

Large Language Models (LLMs) have demonstrated tremendous potential over a wide spectrum of complex reasoning tasks, including legal analysis (Choi et al., 2021; Lai et al., 2024; Fei et al., 2023; Trautmann et al., 2022), robotic manipulation (Driess et al., 2023; Zitkovich et al., 2023; Firoozi et al., 2025; Zhou et al., 2023b), and software development (Anysphere, 2023; Fan et al., 2023; Hou et al., 2024). However, advancing the general reasoning capabilities of LLMs remains a significant challenge (Zhao et al., 2023; Huang & Chang, 2022). To address this, recent breakthroughs in Reinforcement Learning with Verifiable Rewards (RLVR), as exemplified by DeepSeek-R1 (Guo et al., 2025) and OpenAI-o3 (OpenAI, 2025a), have proven that leveraging verifiable rewards as feedback signals can successfully facilitate the emergence of sophisticated reasoning capabilities in LLMs (Lambert et al., 2024; Yang et al., 2025; Kimi et al., 2025).

Despite the encouraging results, conventional RLVR tends to be more applicable to domains with objectively verifiable answers. For instance, in areas such as mathematical proof (Ren et al., 2025; Chen et al., 2025) and code generation (Qwen, 2025; Le et al., 2022), correctness can be explicitly determined through formal proof verification or automated unit tests. In these contexts, the reward signal is clear and well-aligned with the task objective, allowing RLVR to effectively guide models toward correct solutions. Unfortunately, many real-world tasks like medical consultation (Lin et al., 2025; Singhal et al., 2023; Zhang et al., 2023) and creative writing (Wu et al., 2025c; Franceschelli & Musolesi, 2024) are inherently open-ended. These tasks often require multidimensional evaluation and lack a single, verifiable ground-truth answer. To tackle this problem, several very recent

054
055
056
057
058
059
060
061
062
063
064
065
066
067
068
069
070
071
072
073
074
075
076
077
078
079
080
081
082
083
084
085
086
087
088
089
090
091
092
093
094
095
096
097
098
099
100
101
102
103
104
105
106
107

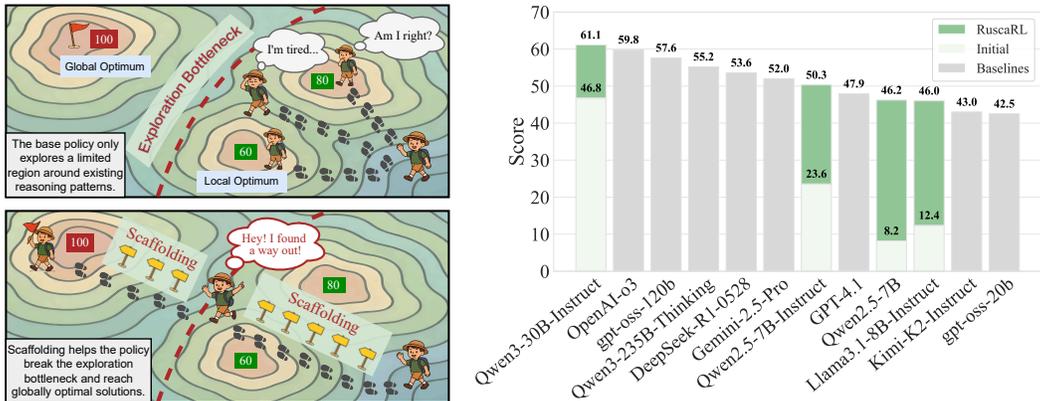


Figure 1: (Left) A conceptual illustration of exploration bottleneck and scaffolding. (Right) Performance comparison of different LLMs on HealthBench-500. Note that Qwen3-30B-Instruct denotes Qwen3-30B-A3B-Instruct, and Qwen3-235B-Thinking denotes Qwen3-235B-A22B-Thinking.

concurrent works (Arora et al., 2025; Kimi et al., 2025; Gunjal et al., 2025; Viswanathan et al., 2025; Huang et al., 2025; Dou et al., 2025) have explored rubric-based evaluation that decomposes desirable responses into checklist-style criteria (e.g., factuality, coherence, completeness). By leveraging LLM-as-a-Judge to score each criterion and aggregating results into scalar rewards, rubrics provide more stable and reliable feedback signals suitable for RLVR in open-ended domains.

Nevertheless, as shown in the left side of Figure 1, a fundamental exploration bottleneck remains as RL requires high-quality samples to improve, yet exploration for such samples remains bounded by the inherent capabilities of LLMs (Yue et al., 2025; Wu et al., 2025a; Liu et al., 2025b; Dong et al., 2025). This creates an inevitable loop where the inability to explore restricts the ability to learn. A growing line of work has attempted to enhance exploration in RLVR for LLMs (Liu et al., 2025b;a; Dong et al., 2025; Zheng et al., 2025; Lei et al., 2025; Li et al., 2025; Cheng et al., 2025). However, these methods largely bias the policy distribution toward high-reward responses already supported by base models, rather than truly expanding its reasoning boundaries (Wu et al., 2025a). Worse still, RL itself has a natural tendency to narrow the exploration space: policy entropy gradually collapses during training, causing the model to converge toward a limited set of reasoning trajectories (Zhao et al., 2025; Yue et al., 2025; Wu et al., 2025a; Yu et al., 2025; Liu et al., 2025b). This, in turn, undermines the potential of RLVR to explore more diverse and higher-quality solutions.

In this work, we introduce Rubric-Scaffolded Reinforcement Learning, termed as RuscaRL, which employs a novel instructional scaffolding framework to break the exploration bottleneck of RLVR. Technically, RuscaRL leverages rubrics in two complementary ways: (1) *Explicit scaffolding during rollout generation*. For each instruction, RuscaRL generates a group of candidate responses by using rubrics as external guidance. We propose intra-group scaffolding differentiation to provide varying levels of rubrics within each group, enabling diverse and high-quality responses. To further internalize underlying reasoning patterns, we use inter-step scaffolding decay to gradually remove the scaffolding over training, thereby minimizing reliance on external guidance. (2) *verifiable rewards during model training*. The model responses are assessed based on multiple criteria derived from rubrics. For each criterion, we employ a Grader LLM that performs a binary evaluation (i.e., true or false), determining whether the response satisfies that specific requirement. The outcomes are then combined through aggregation to obtain a robust reward signal, facilitating effective RL across different general tasks. Our main contributions are summarized as follows:

- We introduce *instructional scaffolding* as a novel paradigm in RLVR for LLMs, which pioneers the integration of external guidance within task instructions to improve rollout diversity and quality, thereby enabling more efficient exploration during RL.
- We propose Rubric-Scaffolded Reinforcement Learning (RuscaRL), an innovative RLVR framework designed to break the exploration bottleneck, integrating checklist-style rubrics as both explicit scaffolding for *exploration* and verifiable rewards for *exploitation*.
- Extensive experiments demonstrate that RuscaRL yields results superior to state-of-the-art counterparts. Notably, RuscaRL enables small LLMs (e.g., Qwen3-30B) to achieve performance on par with leading LLMs (e.g., OpenAI-o3) on HealthBench-500, as shown in the right side of Figure 1.

2 RELATED WORKS

LLM Reasoning. While early methods like prompt engineering (Wei et al., 2022; Kojima et al., 2022; Zhou et al., 2023a; Yao et al., 2023b;a) and supervised fine-tuning (Ouyang et al., 2022) have yielded encouraging results, their reliance on task-specific prompts or extensive labeled data limits their scalability and cross-domain generalization (Stiennon et al., 2020; Pornprasit & Tantithamthavorn, 2024; Zhang et al., 2024b; Gao et al., 2023). Recent works have found that using more test-time computation (Snell et al., 2024; Zhang et al., 2024a; Zuo et al., 2025) can improve the reasoning performance of LLMs. More recently, RLVR (Lambert et al., 2024; OpenAI, 2025a; Guo et al., 2025) has emerged as a promising paradigm for training LLMs to solve verifiable problems, showing strong reasoning improvements in domains like math and coding (Guo et al., 2025; Qwen, 2025; Lambert et al., 2024; OpenAI, 2025a). However, it faces a significant exploration bottleneck (Wu et al., 2025a; Yue et al., 2025; Liu et al., 2025b) and is difficult to extend to general tasks where correctness is hard to verify (Gunjal et al., 2025; Kimi et al., 2025).

Rubric-based Methods. Rubrics are structured evaluation frameworks that decompose complex assessment tasks into specific, verifiable criteria. To address general task evaluation, rubric-based evaluation approaches have emerged across medical (Arora et al., 2025), code (Pathak et al., 2025), and other domains (Fan et al., 2025; Galvan-Sosa et al., 2025; Winata et al., 2025). From the RL perspective, such rubric-based scores can be viewed as a form of reward shaping that grounds learning in structured intermediate feedback and yields denser, more reliable, and more robust reward signals (Ng et al., 1999; Randløv & Alstrøm, 1998; Chan et al., 2024). Building upon these frameworks, researchers apply rubrics as reward signals in RL (Kimi et al., 2025; Gunjal et al., 2025; Viswanathan et al., 2025), using LLMs as graders to provide fine-grained feedback for tasks lacking ground truth. This approach has shown promising results across LLM alignment (Dineen et al., 2025), instruction following (Viswanathan et al., 2025), and open-ended QA (Gunjal et al., 2025; Huang et al., 2025; Dou et al., 2025; Kimi et al., 2025).

Exploration in RL for LLMs. Existing RL methods face insufficient exploration in complex reasoning tasks, with policies trapped in local optima and reasoning boundaries collapsing (Wu et al., 2025a; Yue et al., 2025; Liu et al., 2025b). Current solutions include prolonged training (Liu et al., 2025b;a), entropy-based exploration (Dong et al., 2025; Zheng et al., 2025; Lei et al., 2025; Li et al., 2025; Cheng et al., 2025), curriculum RL (Bengio et al., 2009; Zhang et al., 2025c; Parashar et al., 2025; Gao et al., 2025), and external guidance (Zhang et al., 2025a; RRV et al., 2025), but these approaches fail to break the exploration bottleneck because they either explore within the initial policy distribution or provide only coarse directional signals without structured intermediate guidance. In particular, compared with MeRF (Zhang et al., 2025a), which merely injects a high-level reward description into the prompt as coarse motivational guidance, RuscaRL supplies checklist-style rubrics that decompose the task into fine-grained, verifiable criteria and explicitly scaffold the reasoning trajectory during rollouts. Curriculum learning (Bengio et al., 2009; Parashar et al., 2025; Li et al., 2021; Surkov et al., 2022; Kim & Lee, 2024) reshapes the task distribution by scheduling easier examples before harder ones based on assumed difficulty, whereas RuscaRL keeps the task distribution fixed and instead applies a curriculum only over rubric visibility, gradually decaying checklist scaffolding within each group to enable the policy to internalize these high-quality reasoning patterns.

3 PRELIMINARY

RL Algorithms for LLMs. We adopt Group Relative Policy Optimization (GRPO) (Shao et al., 2024) as our core RL algorithm for training language models with rubric-based rewards. Unlike Proximal Policy Optimization (PPO) (Schulman et al., 2017), GRPO eliminates the need for a value model by using group-based advantage estimation. For each instruction $q \sim \mathcal{D}$, where \mathcal{D} denotes the distribution over the training dataset \mathcal{D} , GRPO samples a group of G responses $\{o_1, o_2, \dots, o_G\}$ from the old policy $\pi_{\theta_{\text{old}}}$ and optimizes the policy π_{θ} by maximizing the following objective:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{q \sim \mathcal{D}, \{o_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot|q)} \left[\frac{1}{G} \sum_{i=1}^G \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \min \left(\rho_{i,t}(\theta) \hat{A}_i, \text{clip}(\rho_{i,t}(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_i \right) \right], \quad (1)$$

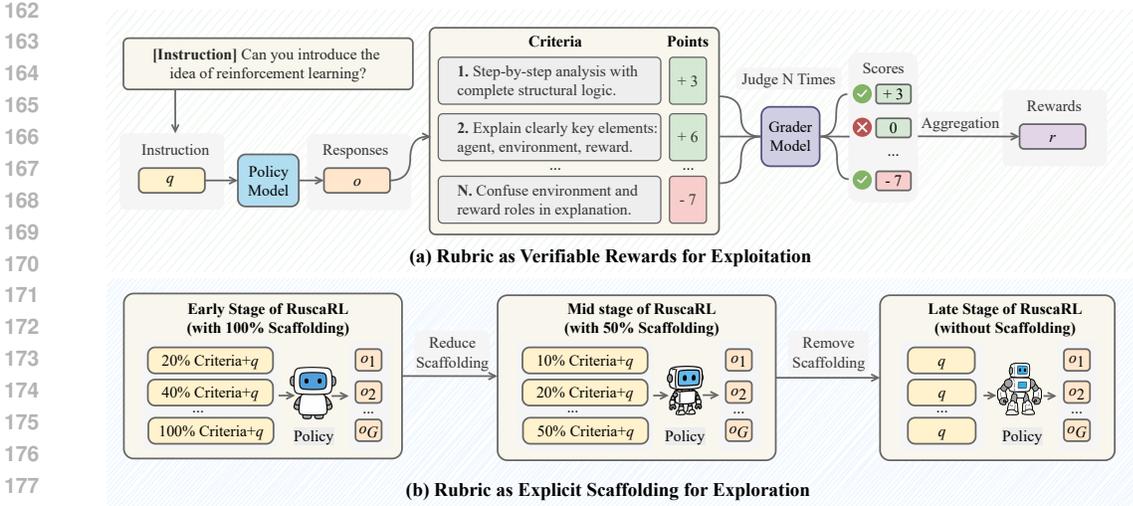


Figure 2: Overview of the RuscaRL framework. (a) Rubric as Verifiable Rewards for Exploitation: checklist-style rubric criteria with associated points are used by an LLM-based grader to perform criterion-wise binary judgments and aggregate the resulting scores into a scalar reward for each response. (b) Rubric as Explicit Scaffolding for Exploration: the same rubrics are injected into instructions as external guidance during rollout generation, with intra-group scaffolding differentiation providing different levels of criteria to samples within a group and inter-step scaffolding decay gradually reducing the amount of scaffolding over training.

where o_i is a response sampled from the old policy $\pi_{\theta_{\text{old}}}$ given instruction q , t denotes the token position within response o_i , $\rho_{i,t}(\theta) = \frac{\pi_{\theta}(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}|q, o_{i,<t})}$ is the token-level importance ratio between the current policy and the previous policy, and ϵ is the clipping coefficient (Schulman et al., 2017). The group-relative advantage is computed as:

$$\hat{A}_i = \frac{r_i - \text{mean}(\{r_j\}_{j=1}^G)}{\text{std}(\{r_j\}_{j=1}^G)}, \quad (2)$$

where r_i is the reward for response o_i , and the advantage is normalized using the mean and standard deviation of the G sampled rewards.

4 METHODOLOGY

To address the exploration bottleneck problem, we propose the RuscaRL framework, as illustrated in Figure 2. RuscaRL leverages rubrics in two complementary ways: (1) *Explicit scaffolding during rollout generation*, where the model generates candidate responses using rubrics as external guidance with intra-group differentiation and inter-step decay (Section 4.2); (2) *Verifiable rewards during model training*, where responses are assessed based on multiple criteria derived from rubrics through binary evaluation and aggregation (Section 4.3). In the following, we first introduce the basic concept of rubrics, then detail these two core components.

4.1 RUBRIC-BASED EVALUATION SYSTEM

Evaluating complex and open-ended tasks is inherently challenging, as responses often differ in structure, style, and content, making it difficult for rule-based evaluation to provide reliable assessments. To address this gap, recent work (Arora et al., 2025) has proposed rubric-based evaluation, which specifies fine-grained, multi-dimensional criteria that can be applied at scale. This design combines the objectivity of automatic metrics with the principled guidance of structured standards, yielding robust scores that better capture response quality, coherence, and completeness.

Formally, a rubric $\mathcal{R} = \{c_1, c_2, \dots, c_N\}$ is defined as a set of N verifiable criteria. Each criterion c_i is specified by a clear description and corresponding points p_i indicating its contribution to the

overall evaluation. We define the point vector as $\mathbf{p} = [p_1, p_2, \dots, p_N]$. For example, given the instruction “Can you introduce the idea of reinforcement learning?”, criteria may include: “step-by-step analysis with complete structural logic” (+3 points), “explain key elements: agent, environment, reward” (+6 points), and negative items like “confuse environment and reward roles in explanation” (-7 points). Points are added or subtracted depending on whether each criterion is satisfied.

Given an instruction q and its corresponding rubric \mathcal{R} both sampled from the data distribution \mathcal{D} , and a model response o generated through the policy model $\pi_\theta(o|q)$, we first construct a judge prompt for each criterion c_i by combining the instruction q , response o , and criterion c_i . The judge prompt template for the grader is provided in Appendix D.1. For a single criterion evaluation, the grader function \mathcal{G} implemented by a LLM (Zheng et al., 2023b; Gu et al., 2024) takes the judge prompt as input and outputs a binary decision $b_i = \mathcal{G}(q, o, c_i) \in \{0, 1\}$ indicating whether criterion c_i is satisfied (true or false). Extending this to the full rubric, the grader evaluates all criteria and produces a binary indicator vector $\mathbf{b} = \mathcal{G}(q, o, \mathcal{R}) = [b_1, b_2, \dots, b_N]$ where each b_i represents the satisfaction of criterion c_i . The final score vector is obtained by element-wise multiplication: $\mathbf{s} = \mathbf{b} \odot \mathbf{p} = [b_1 p_1, b_2 p_2, \dots, b_N p_N]$, providing fine-grained score across all specified criteria. We also compute the total possible score $S_{total} = \sum_{j=1}^M p_j$ where M is the number of positive-point criteria, which will be used for normalization in the reward calculation.

4.2 RUBRIC-BASED SCAFFOLDING MECHANISM FOR RL EXPLORATION

During RL training on complex reasoning tasks, models often fail to sustain effective exploration (Wu et al., 2025a; Yue et al., 2025; Liu et al., 2025b): initial randomness quickly diminishes, policy entropy collapses, and the model prematurely converges to suboptimal reasoning patterns. This collapse severely limits the discovery of diverse and high-quality solution trajectories.

To mitigate this issue, we draw inspiration from instructional scaffolding theory in educational psychology (Vygotsky & Cole, 1978). According to Vygotsky’s Zone of Proximal Development, when learners’ capabilities are insufficient, they benefit from structured guidance to bridge the gap between current ability and target performance. As competence grows, this scaffolding should be gradually withdrawn to foster independent problem-solving (Wood et al., 1976).

Building on this insight, we design a rubric-based scaffolding mechanism that provides varying numbers of rubric criteria as explicit guidance throughout the training process, helping models gradually learn to generate high-quality responses. Specifically, our rubric-based scaffolding mechanism augments the original policy function by adding a subset of rubric criteria \mathcal{R}_S as additional guidance, representing the policy as $\pi_\theta(o|q, \mathcal{R}_S)$. The specific prompt template for incorporating scaffolding is detailed in Appendix D.2. Additionally, we design a two-dimensional control mechanism to determine the rubrics scaffolding ratio λ_S , and then sample criteria from the complete rubric set \mathcal{R} to form \mathcal{R}_S , *i.e.*, $|\mathcal{R}_S| = \text{round}(\lambda_S \times |\mathcal{R}|)$. We instantiate this mechanism in two dimensions: intra-group Scaffolding differentiation and inter-step Scaffolding decay.

Intra-Group Scaffolding Differentiation. In RL algorithms with multi-sampling, such as GRPO, computing group-relative advantages (Eq. 2) requires response diversity to avoid collapse into homogeneous samples. To this end, we assign different levels of rubric scaffolding within each group, encouraging both guided and independent exploration. Concretely, we define a group-level ratio vector $\lambda_{group} = [\lambda_1, \lambda_2, \dots, \lambda_G]$ where $\lambda_i = \frac{G-i}{G-1}$ for the i -th sample in the group of size G . This linear differentiation ensures that some samples benefit from stronger scaffolding while others are deliberately exposed to weaker guidance, thereby enhancing intra-group diversity.

Inter-Step Scaffolding Decay. Inspired by instructional scaffolding theory, we gradually reduce guidance as the model develops independent learning strategies using a sigmoid function $\lambda_{step}(t) = \frac{1}{1+e^{\alpha(t-t_0)}}$, where t is the current training progress ($t \in [0, 1]$), t_0 is the midpoint, and α controls the steepness of decay. This mechanism prevents overfitting to external guidance by creating an adaptive learning environment where the model initially benefits from guidance to overcome the exploration bottleneck, then gradually transitions to independent reasoning as capabilities mature.

Integrated Scaffolding Mechanism. Finally, we combine intra-group differentiation and inter-step decay into a unified ratio vector:

$$\lambda_S = \lambda_{step}(t) \times \lambda_{group} = [\lambda_{S,1}, \lambda_{S,2}, \dots, \lambda_{S,G}], \quad (3)$$

where $\lambda_{S,i} = \lambda_{step}(t) \times \lambda_i = \frac{1}{1+e^{\alpha(t-t_0)}} \times \frac{G-i}{G-1}$ represents the scaffolding ratio for the i -th sample in the group. This integrated mechanism simultaneously promotes response diversity within each group and adaptively reduces reliance on scaffolding across training steps, jointly addressing the problems of homogeneity and overfitting.

4.3 RUBRIC-BASED REWARD FUNCTION FOR RL EXPLOITATION

To provide robust and reliable reward signals for general reasoning tasks, we design rubric-based reward functions. The multi-dimensional score vector $\mathbf{s} = [s_1, s_2, \dots, s_N]$ obtained from the evaluation system is aggregated into a final scalar reward by directly summing all criterion scores and normalizing by the total possible score computed in Section 4.1:

$$S = \frac{\sum_{i=1}^N s_i}{S_{total}}, \quad (4)$$

where S represents the final score, s_i is the score of the i -th criterion, and S_{total} is the total possible score of all positive-point criteria computed in Section 4.1. This calculation method results in scores that fall within the interval $[0, 1]$ in most cases, with occasional negative scores possible. We directly adopt this rubric-based score S as our reward: $r_i = S_i$, where r_i is the reward for the i -th response. This approach enables application to open-ended tasks without ground truth answers while providing more robust assessment than holistic LLM scoring.

Once the rubric-based rewards are obtained, we employ them to train the policy model using RL algorithms. The training process follows the policy gradient framework, where the model learns to maximize the expected reward. Algorithm 1 in Appendix A outlines the complete training procedure. Additionally, to help the model better internalize underlying reasoning patterns, the log probability computation in training is based on $\pi_{\theta}(o_{i,t}|q, o_{i,<t})$ rather than $\pi_{\theta}(o_{i,t}|q, \mathcal{R}_S, o_{i,<t})$. For detailed analysis on importance sampling, see Appendix C.5.

5 EXPERIMENTS

To demonstrate the effectiveness of the proposed RuscaRL method, we conduct experiments across multiple benchmarks spanning medical, writing, instruction following, and STEM domains. Our experiments seek to answer the following questions: (1) Does RuscaRL demonstrate consistent effectiveness across different models and tasks, and how does it compare against existing fine-tuning methods? (Section 5.2 and Appendix C.1, C.2, C.3) (2) How does RuscaRL break the exploration bottleneck in RL for LLM reasoning? (Section 5.3 and Appendix C.4) (3) What is the impact of different components in the rubric-based scaffolding mechanism? (Section 5.4 and Appendix C.5)

5.1 EXPERIMENTAL SETUPS

Models and Training Settings. We use multiple initial models from different series and parameter scales for our experiments, including both instruct models and base models from the Qwen2.5 series (Yang et al., 2024), the Qwen3 series (Yang et al., 2025), and the Llama-3 series (Meta-AI, 2025; Grattafiori et al., 2024). All models are trained using the GRPO algorithm on the verl framework (Sheng et al., 2025). Detailed training settings are provided in Appendix B.1.

Evaluation Benchmarks. We use HealthBench-500, a randomly selected subset of 500 samples from HealthBench (Arora et al., 2025), as our held-out evaluation set. Additionally, we evaluate on other medical benchmarks including LLMEval-Med (Zhang et al., 2025b), MedQA (Jin et al., 2021), and MedMCQA (Pal et al., 2022). For the Writing domain, we use WritingBench (Wu et al., 2025c) and Creative Writing v3 (Paech, 2025) benchmarks. For the Instruction Following domain, we use IFEVAL (Zhou et al., 2023c) and IFBench (Pyatkin et al., 2025) benchmarks. For the STEM domain, we use GPQA-Diamond (Rein et al., 2024), MMLU (Hendrycks et al., 2020), MMLU-Pro (Wang et al., 2024), MATH-500 (Lightman et al., 2023), AMC 2023¹, AIME 2024, and AIME 2025². Detailed evaluation settings are provided in Appendix B.2.

¹<https://huggingface.co/datasets/knoveleng/AMC-23>

²https://artofproblemsolving.com/wiki/index.php/AIME_Problems_and_Solutions

Table 1: Main results comparison with closed-source and open-source models across different benchmarks. The best results in each box are highlighted in **bold**.

Model	Medical				Writing		Instruction Following	
	HealthBench-500	LLMEval-Med	MedQA	MedMCQA	WritingBench	Creative Writing	IFEVAL	IFBench
Closed-source and Open-source Models								
OpenAI-o3	59.8	78.9	96.0	84.7	77.7	81.4	91.6	67.8
GPT-4.1	47.9	71.2	92.4	80.0	69.0	79.0	87.0	37.2
gpt-oss-20b	42.5	68.8	85.6	68.1	66.6	39.4	83.5	48.7
Rubicon-Preview	50.8	73.2	85.1	70.7	73.0	66.4	82.4	33.4
Our Models								
Qwen3-30B-A3B-Instruct	46.8	71.4	84.2	71.3	78.4	74.4	83.0	31.9
+ RuscaRL	61.1 (+14.3)	73.0 (+1.6)	84.8 (+0.6)	71.9 (+0.6)	79.2 (+0.8)	74.3 (-0.1)	84.5 (+1.5)	32.1 (+0.2)
Qwen3-30B-A3B-Base	11.4	43.3	73.6	65.1	36.8	35.8	39.0	13.3
+ RuscaRL	48.3 (+36.9)	60.9 (+17.6)	71.3 (-2.3)	65.4 (+0.3)	59.0 (+22.2)	46.0 (+10.2)	76.3 (+37.3)	30.3 (+17.0)
Qwen2.5-7B-Instruct	23.6	47.9	61.8	56.3	45.3	37.4	71.0	26.8
+ RuscaRL	50.3 (+26.7)	61.2 (+13.3)	63.5 (+1.7)	56.5 (+0.2)	56.3 (+11.0)	38.6 (+1.2)	75.3 (+4.3)	31.0 (+4.2)
Qwen2.5-7B	8.2	28.2	55.3	55.0	23.5	30.3	32.0	14.5
+ RuscaRL	46.2 (+38.0)	47.8 (+19.6)	58.2 (+2.9)	55.6 (+0.6)	45.8 (+22.3)	34.8 (+4.5)	56.2 (+24.2)	25.9 (+11.4)
Llama-3.1-8B-Instruct	12.4	29.8	66.8	58.0	36.7	44.5	72.6	22.6
+ RuscaRL	46.0 (+33.6)	46.3 (+16.5)	70.7 (+3.9)	60.7 (+2.7)	52.7 (+16.0)	54.2 (+9.7)	79.7 (+7.1)	31.1 (+8.5)
Llama-3.1-8B	0	9.1	36.9	35.9	12.9	26.3	18.1	11.6
+ RuscaRL	25.8 (+25.8)	29.5 (+20.4)	49.7 (+12.8)	45.4 (+9.5)	35.6 (+22.7)	33.3 (+7.0)	55.6 (+37.5)	21.4 (+9.8)

Baselines. We compare RuscaRL against four representative baseline methods: (1) Rubric-based RL: A rubric-as-reward RL ablation baseline implemented with the GRPO algorithm using rubric scores as rewards, corresponding to Rubric as Verifiable Rewards for Exploitation without scaffolding mechanisms. (2) Rubric-based RL with full scaffolding (Rubric-based RL-S): A method that provides all rubrics with scaffolding support in the instruction, without intra-group differentiation and without an inter-step decay function. (3) SFT: Fine-tuned on GPT-4.1 (OpenAI, 2025b) demonstrations generated with scaffolding support. (4) SFT + Rubric-based RL: A combination approach that first applies SFT and then applies rubric-based RL training.

5.2 OVERALL PERFORMANCE

RuscaRL achieves consistent and notable gains across tasks and model scales, showcasing its effectiveness and broad generalization. Across medical, writing, and instruction-following tasks (Table 1), RuscaRL achieves significant gains over multiple initial models, with Qwen3-30B-A3B-Instruct on HealthBench-500 surpassing many leading models (e.g., OpenAI-o3). It is worth noting that our training data consists entirely of open-ended tasks, whereas MedQA and MedMCQA are closed-ended multiple-choice benchmarks. Our improvements on these closed-ended benchmarks are marginal and are included only to demonstrate cross-task generalization rather than to claim major gains. Notably, RuscaRL is particularly effective on instruct models and provides larger benefits for weaker models, such as Llama-3.1-8B-Instruct. This advantage stems from our scaffolding approach, which leverages the model’s existing instruction-following ability to elicit higher-quality and more diverse responses, explaining why RuscaRL is especially well-suited for training on models with strong instruction-following capacities. Meanwhile, RuscaRL has also been successfully extended to the STEM domain: experiments on Qwen2.5-7B-Instruct show consistent performance improvements across all STEM benchmarks (see Figure 3). More detailed results about performance across different model series and scales are provided in Appendix C.1, which further demonstrates the robustness and broad applicability of our approach. Additionally, we explore the effects of mixing training data from different domains in Appendix C.2.

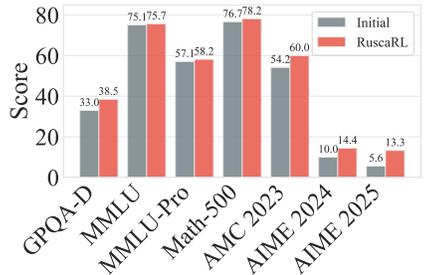


Figure 3: STEM Benchmarks.

RuscaRL consistently outperforms Rubrics-based methods across tasks. As shown in Table 2, in the **direct RL** setting, RuscaRL achieves the best performance on most medical, writing, and instruction-following tasks, delivering larger and more stable gains than Rubric-based RL and RL-S (e.g., 50.3 vs. 41.2 and 36.6 accuracy on HealthBench-500 with Qwen2.5-7B-Instruct). In the **SFT-then-RL** setting, both RuscaRL and Rubric-based RL achieve additional improvements over SFT,

Table 2: Performance comparison of different baseline methods and RuscaRL across multiple domains. **Bold** indicates the best performance of each method.

Method	Medical				Writing		Instruction Following	
	HealthBench-500	LLMEval-Med	MedQA	MedMCQA	WritingBench	Creative Writing	IFEVAL	IFBench
Qwen2.5-7B-Instruct								
Initial	23.6	47.9	61.8	56.3	45.3	37.4	71.0	26.8
Rubric-based RL	41.2 (+17.6)	54.6 (+6.7)	62.1 (+0.3)	56.3 (+0.0)	53.7 (+8.4)	38.8 (+1.4)	75.1 (+4.1)	29.3 (+2.5)
Rubric-based RL-S	36.6 (+13.0)	56.1 (+8.2)	57.9 (-3.9)	52.4 (-3.9)	45.8 (+0.5)	38.3 (+0.9)	71.9 (+0.9)	28.6 (+1.8)
RuscaRL (Ours)	50.3 (+26.7)	61.2 (+13.3)	63.5 (+1.7)	56.5 (+0.2)	56.3 (+11.0)	38.6 (+1.2)	75.3 (+4.3)	31.0 (+4.2)
SFT	38.3 (+14.7)	52.6 (+4.7)	60.8 (-1.0)	57.3 (+1.0)	62.8 (+17.5)	45.3 (+7.9)	75.2 (+4.2)	25.2 (-1.6)
SFT + Rubric-based RL	55.1 (+31.5)	58.5 (+10.6)	59.7 (-2.1)	56.4 (+0.1)	66.7 (+21.4)	43.6 (+6.2)	82.1 (+11.1)	29.6 (+2.8)
SFT + RuscaRL (Ours)	56.9 (+33.3)	58.8 (+10.9)	61.6 (-0.2)	56.9 (+0.6)	66.8 (+21.5)	43.9 (+6.5)	82.5 (+11.5)	30.6 (+3.8)
Qwen2.5-7B								
Initial	8.2	28.2	55.3	55.0	23.5	30.3	32.0	14.5
Rubric-based RL	41.9 (+33.7)	46.5 (+18.3)	48.2 (-7.1)	49.9 (-5.1)	40.2 (+16.7)	33.8 (+3.5)	53.4 (+21.4)	25.5 (+11.0)
Rubric-based RL-S	21.7 (+13.5)	44.4 (+16.2)	60.3 (+5.0)	55.5 (+0.5)	43.9 (+20.4)	25.7 (-4.6)	52.3 (+20.3)	20.4 (+5.9)
RuscaRL (Ours)	46.2 (+38.0)	47.8 (+19.6)	58.2 (+2.9)	55.6 (+0.6)	45.8 (+22.3)	34.8 (+4.5)	56.2 (+24.2)	25.9 (+11.4)
SFT	32.2 (+24.0)	40.0 (+11.8)	56.5 (+1.2)	54.4 (-0.6)	56.6 (+33.1)	42.5 (+12.2)	69.7 (+37.7)	20.8 (+6.3)
SFT + Rubric-based RL	36.4 (+28.2)	39.7 (+11.5)	57.1 (+1.8)	54.1 (-0.9)	57.1 (+33.6)	43.2 (+12.9)	71.6 (+39.6)	23.7 (+9.2)
SFT + RuscaRL (Ours)	35.4 (+27.2)	42.7 (+14.5)	58.2 (+2.9)	55.1 (+0.1)	57.4 (+33.9)	42.6 (+12.3)	72.0 (+40.0)	23.1 (+8.6)

with RuscaRL generally yielding larger gains across most tasks, though the magnitude is smaller than in the direct RL setting. We argue that RuscaRL essentially leverages rubrics as prior knowledge to guide exploration, while SFT also serves to accelerate RL exploration (cold start). Since both mechanisms overlap in facilitating exploration, this may explain why the performance gap between RuscaRL and Rubric-based RL narrows under the SFT-then-RL setting.

5.3 ANALYSIS

In this subsection, we conduct an analysis of RuscaRL using Qwen2.5-7B-Instruct as the initial model and HealthBench as the training and evaluation dataset. In addition, we compare three approaches in the following analysis: RuscaRL, RuscaRL* (RuscaRL without the inter-step decay mechanism), and Rubric-based RL. We use the Best-of-N³ metric to reflect both the model’s reasoning boundaries (at large N) and sampling efficiency (at small N).

RuscaRL significantly improves sampling efficiency and reasoning boundaries. As shown in Figure 4, RuscaRL significantly improves single-sample quality at N=1, indicating that the scaffolding mechanism effectively enhances the model’s reasoning stability. At N=2048, its performance ceiling surpasses both the initial model and Rubric-based RL, validating its advantage in expanding the reasoning boundary. Moreover, RuscaRL exhibits a steeper performance curve across N, meaning it can achieve the same performance with fewer samples. Further analysis (Appendix C.4.1) indicates that RuscaRL also produces highly novel responses that the initial model could barely generate, showing that rubric scaffolding effectively breaks the exploration bottleneck and discovers new solutions.

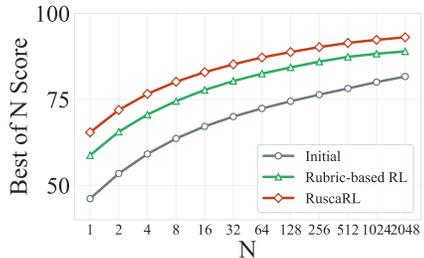


Figure 4: Best-of-N Performance.

RuscaRL achieves exploration-exploitation balance. As shown in Figure 5a, RuscaRL demonstrates a well-balanced exploration–exploitation trajectory: policy entropy first rises as the model explores diverse reasoning trajectories, then declines as it converges to high-quality patterns. In contrast, RuscaRL* suffers from uncontrolled entropy growth leading to instability, while Rubric-based RL collapses under continuous entropy decline. Validation accuracy (Figure 5b) consistently shows RuscaRL achieving the best performance throughout training, demonstrating long-term stability without policy entropy collapses, followed by Rubric-based RL and then RuscaRL*. Similar trends are observed in Self-BLEU and Semantic Distance (Appendix C.4.2), confirming that RuscaRL achieves effective exploration followed by stable exploitation.

³For cost considerations in the Best-of-N evaluation, we use Qwen3-32B (non-thinking) as the Grader LLM.

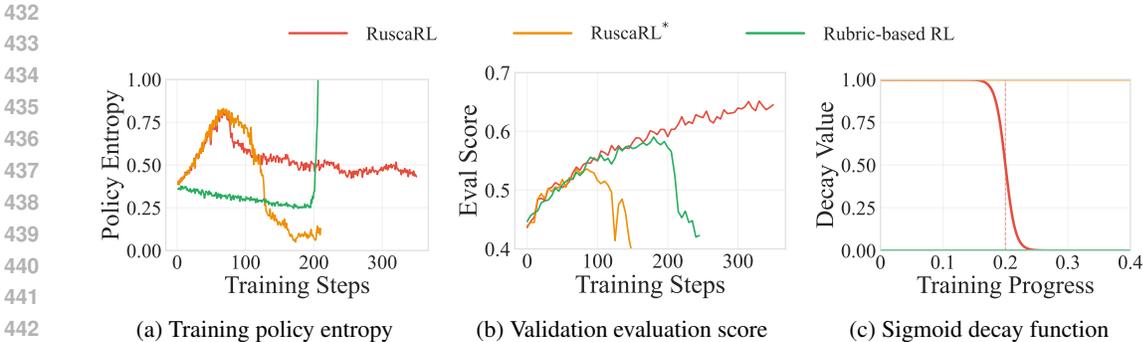


Figure 5: Training dynamics. The figure shows the evolution of policy entropy, validation accuracy, and sigmoid decay function during training.

5.4 ABLATION STUDIES

Intra-group Differentiation Analysis. We first analyze different strategies for the intra-group control mechanism using Qwen2.5-7B-Instruct as the initial model and HealthBench as the training and evaluation dataset. Within individual sampling groups, we compare different rubric scaffolding differentiation patterns. These mechanisms are: **(1) Linear (Ours):** Linear differentiation pattern following our proposed formula $\frac{G-i}{G-1}$, providing different levels of rubric scaffolding to different samples within a single sampling group. **(2) Binary:** Binary differentiation patterns where N represents the number of samples with full rubric scaffolding within a single sampling group, including configurations such as no-scaffolding ($N=0$), half-scaffolding ($N=4$), and full-scaffolding ($N=8$).

As shown in Figure 6a, the linear differentiation strategy performs optimally in intra-group control. This result can be attributed to the linear strategy’s significant enhancement of sampling diversity, which works synergistically with multi-sampling algorithms like GRPO.

Inter-step Decay Analysis. We analyze different decay functions for inter-step control during training. We define the base scaffolding intensity of inter-step control as $f(t)$, where t is the normalized training progress ($t \in [0, 1]$). We compare the following decay functions: **(1) Sigmoid (Ours):** S-shaped decay function $f(t) = \frac{1}{1+e^{\alpha(t-t_0)}}$, where parameter α controls the steepness of decay and t_0 controls the midpoint of decay, achieving smooth nonlinear transitions. **(2) Constant:** Constant control $f(t) = 1$, maintaining constant full scaffolding. **(3) Linear:** Linear decay function $f(t) = 1-t$, achieving uniform linear decrease. **(4) Power (n):** Power decay function $f(t) = (1-t)^n$, where n controls the curvature of decay, including various power configurations.

As shown in Figure 6b, the sigmoid decay function achieves the best performance among all decay strategies. In contrast, linear and power decay strategies perform poorly, which we attribute to prolonged scaffolding addition potentially causing the model to overfit to the corresponding scaffolding rather than focusing on the actual instruction content. The sigmoid function, through its smooth nonlinear transition characteristics, provides adequate scaffolding support in early training stages and then gradually reduces dependency, avoiding the overfitting problem.

Based on the superior performance of the sigmoid function, we further analyze the effects of both parameter dimensions (speed α and midpoint t_0) using Qwen2.5-7B-Instruct as the initial model and HealthBench as the training and evaluation dataset. Figures 6c and 6d demonstrate the performance differences across various sigmoid parameter configurations, ultimately determining the optimal configuration as $\alpha = 125, t_0 = 0.2$. (1) Removing scaffolding too fast (large α) prevents the model from adapting to rapid scaffolding changes, easily causing training instability; while removing scaffolding too slowly (small α) leads to incomplete early-stage scaffolding, failing to fully stimulate the model’s exploration capability, and prolonged retention of scaffolding in later stages also causes overfitting issues. (2) Starting decay too early (small t_0) leads to insufficient scaffolding support, causing the model to lack necessary guidance in early training stages; while starting decay too late (large t_0) causes the model to over-rely on scaffolding, ultimately resulting in overfitting.

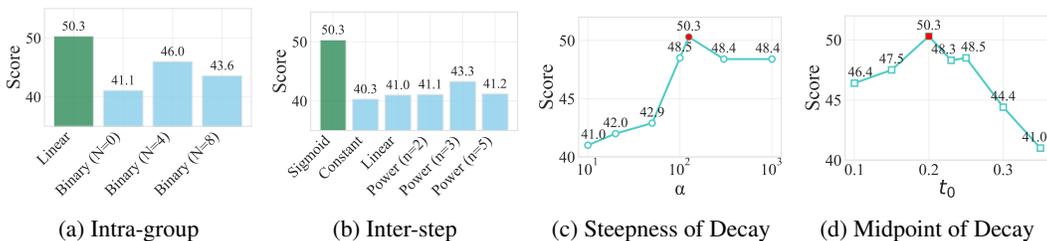


Figure 6: Ablation studies on RuscaRL framework components. (a) Intra-group differentiation strategies comparison; (b) Inter-step decay functions comparison; (c) Sigmoid parameter steepness of decay α sensitivity analysis with fixed $t_0 = 0.2$; (d) Sigmoid parameter midpoint of decay t_0 sensitivity analysis with fixed $\alpha = 125$.

6 CONCLUSION AND DISCUSSION

In this work, we apply instructional scaffolding theory from educational psychology to RL for LLMs, and introduce RuscaRL, a novel instructional scaffolding framework that breaks the exploration bottleneck for general LLM reasoning tasks. RuscaRL leverages checklist-style rubrics through scaffolding mechanisms that provide gradually decaying external guidance and reward functions that enable robust RL training. The rubric-based scaffolding mechanism provides external guidance that gradually decays to encourage internalization, while the rubric-based reward function enables robust evaluation for effective RL training. Extensive experiments demonstrate that RuscaRL consistently outperforms strong baseline methods and achieves competitive results compared with leading models. Furthermore, the model fine-tuned with RuscaRL produces highly novel responses that the initial model could barely generate.

While RuscaRL demonstrates promising results in breaking the exploration bottleneck for general LLM reasoning, several limitations remain that highlight directions for future research. Our approach critically relies on high-quality, well-structured rubric datasets, which are still scarce in the community, and is highly sensitive to rubric design quality. Poorly designed rubrics may fail to provide robust reward signals due to unreasonable point allocations or conflicting criteria, while narrow rubrics may limit the scaffolding process’s ability to generate diverse, high-quality responses. The success of RuscaRL underscores the urgent need for broader community investment in constructing open, diverse, and domain-rich rubric datasets. Our future work includes developing pipelines for high-quality rubric data production, exploring rubric-based natural language feedback strategies, and investigating applications to multi-modal tasks and agent systems.

REPRODUCIBILITY STATEMENT

To ensure the reproducibility of our work, we have made efforts to provide the necessary details for replicating our results. Experimental settings and implementation details are documented in Appendix B. The complete prompt templates used in our experiments are provided in Appendix D. Additionally, we include our code as supplementary materials, which contains scripts and configurations needed to reproduce the experimental results presented in this paper.

REFERENCES

- Anysphere. Cursor: The ai code editor, 2023. URL <https://cursor.com/home?from=agents>.
- Rahul K Arora, Jason Wei, Rebecca Soskin Hicks, Preston Bowman, Joaquin Quiñonero-Candela, Foivos Tsimpourlas, Michael Sharman, Meghan Shah, Andrea Vallone, Alex Beutel, et al. Health-bench: Evaluating large language models towards improved human health. *arXiv preprint arXiv:2505.08775*, 2025.

- 540 Yushi Bai, Jiajie Zhang, Xin Lv, Linzhi Zheng, Siqi Zhu, Lei Hou, Yuxiao Dong, Jie Tang, and
541 Juanzi Li. Longwriter: Unleashing 10,000+ word generation from long context llms. *arXiv*
542 *preprint arXiv:2408.07055*, 2024.
- 543
- 544 Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In
545 *ICML*, pp. 41–48, 2009.
- 546 Alex J Chan, Hao Sun, Samuel Holt, and Mihaela Van Der Schaar. Dense reward for free in rein-
547 forcement learning from human feedback. *ICML*, 2024.
- 548
- 549 Luoxin Chen, Jinming Gu, Liankai Huang, Wenhao Huang, Zhicheng Jiang, Allan Jie, Xiaoran Jin,
550 Xing Jin, Chenggang Li, Kaijing Ma, et al. Seed-prover: Deep and broad reasoning for automated
551 theorem proving. *arXiv preprint arXiv:2507.23726*, 2025.
- 552 Daixuan Cheng, Shaohan Huang, Xuekai Zhu, Bo Dai, Wayne Xin Zhao, Zhenliang Zhang, and
553 Furu Wei. Reasoning with exploration: An entropy perspective. *arXiv preprint arXiv:2506.14758*,
554 2025.
- 555
- 556 Jonathan H Choi, Kristin E Hickman, Amy B Monahan, and Daniel Schwarcz. Chatgpt goes to law
557 school. *J. Legal Educ.*, 71:387, 2021.
- 558 Jacob Dineen, Aswin RRV, Qin Liu, Zhikun Xu, Xiao Ye, Ming Shen, Zhaonan Li, Shijie Lu, Chitta
559 Baral, Muhao Chen, et al. Qa-lign: Aligning llms through constitutionally decomposed qa. *arXiv*
560 *preprint arXiv:2506.08123*, 2025.
- 561
- 562 Yihong Dong, Xue Jiang, Yongding Tao, Huanyu Liu, Kechi Zhang, Lili Mou, Rongyu Cao, Ying-
563 wei Ma, Jue Chen, Binhua Li, et al. RL-plus: Countering capability boundary collapse of llms in
564 reinforcement learning with hybrid-policy optimization. *arXiv preprint arXiv:2508.00222*, 2025.
- 565 Chengfeng Dou, Chong Liu, Fan Yang, Fei Li, Jiyuan Jia, Mingyang Chen, Qiang Ju, Shuai Wang,
566 Shunya Dang, Tianpeng Li, et al. Baichuan-m2: Scaling medical capability with large verifier
567 system. *arXiv preprint arXiv:2509.02208*, 2025.
- 568
- 569 Danny Driess, Fei Xia, Mehdi S. M. Sajjadi, Corey Lynch, Aakanksha Chowdhery, Brian Ichter,
570 Ayzaan Wahid, Jonathan Tompson, Quan Vuong, Tianhe Yu, Wenlong Huang, Yevgen Chebotar,
571 Pierre Sermanet, Daniel Duckworth, Sergey Levine, Vincent Vanhoucke, Karol Hausman, Marc
572 Toussaint, Klaus Greff, Andy Zeng, Igor Mordatch, and Pete Florence. Palm-e: an embodied
573 multimodal language model. In *ICML*, 2023.
- 574 Angela Fan, Beliz Gokkaya, Mark Harman, Mitya Lyubarskiy, Shubho Sengupta, Shin Yoo, and
575 Jie M Zhang. Large language models for software engineering: Survey and open problems. In
576 *ICSE-FoSE*, pp. 31–53, 2023.
- 577 Zhiyuan Fan, Weinong Wang, Xing Wu, and Debing Zhang. Sedareval: Automated evaluation using
578 self-adaptive rubrics. *arXiv preprint arXiv:2501.15595*, 2025.
- 579
- 580 Zhiwei Fei, Xiaoyu Shen, Dawei Zhu, Fengzhe Zhou, Zhuo Han, Songyang Zhang, Kai Chen,
581 Zongwen Shen, and Jidong Ge. Lawbench: Benchmarking legal knowledge of large language
582 models. *arXiv preprint arXiv:2309.16289*, 2023.
- 583
- 584 Roya Firoozi, Johnathan Tucker, Stephen Tian, Anirudha Majumdar, Jiankai Sun, Weiyu Liu, Yuke
585 Zhu, Shuran Song, Ashish Kapoor, Karol Hausman, et al. Foundation models in robotics: Appli-
586 cations, challenges, and the future. *IJRR*, 44(5):701–739, 2025.
- 587 Giorgio Franceschelli and Mirco Musolesi. On the creativity of large language models. *AI &*
588 *SOCIETY*, pp. 1–11, 2024.
- 589 Diana Galvan-Sosa, Gabrielle Gaudeau, Pride Kavumba, Yunmeng Li, Zheng Yuan, Keisuke Sak-
590 aguchi, Paula Buttery, et al. Rubrik’s cube: Testing a new rubric for evaluating explanations on
591 the cube dataset. *arXiv preprint arXiv:2503.23899*, 2025.
- 592
- 593 Luyu Gao, Aman Madaan, Shuyan Zhou, Uri Alon, Pengfei Liu, Yiming Yang, Jamie Callan, and
Graham Neubig. Pal: Program-aided language models. In *ICML*, pp. 10764–10799, 2023.

- 594 Zhaolin Gao, Joongwon Kim, Wen Sun, Thorsten Joachims, Sid Wang, Richard Yuanzhe Pang,
595 and Liang Tan. Prompt curriculum learning for efficient llm post-training. *arXiv preprint*
596 *arXiv:2510.01135*, 2025.
- 597 Google. Gemini 2.5 pro: Best for coding and highly complex tasks, 2025. URL [https://](https://deepmind.google/models/gemini/pro/)
598 deepmind.google/models/gemini/pro/.
- 600 Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The llama 3 herd of models. *arXiv*
601 *preprint arXiv:2407.21783*, 2024.
- 602 Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Ying-
603 han Shen, Shengjie Ma, Honghao Liu, et al. A survey on llm-as-a-judge. *arXiv preprint*
604 *arXiv:2411.15594*, 2024.
- 606 Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Bing Liu, and Sean Hendryx. Rubrics as
607 rewards: Reinforcement learning beyond verifiable domains. *arXiv preprint arXiv:2507.17746*,
608 2025.
- 609 Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu,
610 Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms
611 via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.
- 612 Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and
613 Jacob Steinhardt. Measuring massive multitask language understanding. *arXiv preprint*
614 *arXiv:2009.03300*, 2020.
- 616 Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn
617 Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset.
618 In *NeurIPS*, volume 1, 2021.
- 619 Xinyi Hou, Yanjie Zhao, Yue Liu, Zhou Yang, Kailong Wang, Li Li, Xiapu Luo, David Lo, John
620 Grundy, and Haoyu Wang. Large language models for software engineering: A systematic litera-
621 ture review. *ACM TOSEM*, 33(8):1–79, 2024.
- 622 Jie Huang and Kevin Chen-Chuan Chang. Towards reasoning in large language models: A survey.
623 *arXiv preprint arXiv:2212.10403*, 2022.
- 625 Zenan Huang, Yihong Zhuang, Guoshan Lu, Zeyu Qin, Haokai Xu, Tianyu Zhao, Ru Peng, Jiaqi
626 Hu, Zhanming Shen, Xiaomeng Hu, et al. Reinforcement learning with rubric anchors. *arXiv*
627 *preprint arXiv:2508.12790*, 2025.
- 628 Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. What dis-
629 ease does this patient have? a large-scale open domain question answering dataset from medical
630 exams. *Applied Sciences*, 11(14):6421, 2021.
- 631 Jisu Kim and Juhwan Lee. Strategic data ordering: Enhancing large language model performance
632 through curriculum learning. *arXiv preprint arXiv:2405.07490*, 2024.
- 634 Kimi, Yifan Bai, Yiping Bao, Guanduo Chen, Jiahao Chen, Ningxin Chen, Ruijue Chen, Yanru
635 Chen, Yuankun Chen, Yutian Chen, et al. Kimi k2: Open agentic intelligence. *arXiv preprint*
636 *arXiv:2507.20534*, 2025.
- 637 Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large
638 language models are zero-shot reasoners. In *NeurIPS*, volume 35, pp. 22199–22213, 2022.
- 640 Jinqi Lai, Wensheng Gan, Jiayang Wu, Zhenlian Qi, and Philip S Yu. Large language models in law:
641 A survey. *AI Open*, 5:181–196, 2024.
- 642 Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brah-
643 man, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers
644 in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.
- 646 Hung Le, Yue Wang, Akhilesh Deepak Gotmare, Silvio Savarese, and Steven Chu Hong Hoi.
647 Coder1: Mastering code generation through pretrained models and deep reinforcement learning.
In *NeurIPS*, volume 35, pp. 21314–21328, 2022.

- 648 Shiye Lei, Zhihao Cheng, Kai Jia, and Dacheng Tao. Revisiting llm reasoning via information
649 bottleneck. *arXiv preprint arXiv:2507.18391*, 2025.
- 650
- 651 Conglong Li, Minjia Zhang, and Yuxiong He. Curriculum learning: A regularization method for
652 efficient and stable billion-scale gpt model pre-training. *arXiv preprint arXiv:2108.06084*, 8:13,
653 2021.
- 654 Xianzhi Li, Ethan Callanan, Xiaodan Zhu, Mathieu Sibue, Antony Papadimitriou, Mahmoud Mah-
655 fouz, Zhiqiang Ma, and Xiaomo Liu. Entropy-aware branching for improved mathematical rea-
656 soning. *arXiv preprint arXiv:2503.21961*, 2025.
- 657
- 658 Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan
659 Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let’s verify step by step. In *ICLR*, 2023.
- 660 Tianwei Lin, Wenqiao Zhang, Sijing Li, Yuqian Yuan, Binhe Yu, Haoyuan Li, Wanggui He, Hao
661 Jiang, Mengze Li, Xiaohui Song, et al. Healthgpt: A medical large vision-language model for
662 unifying comprehension and generation via heterogeneous knowledge adaptation. *arXiv preprint*
663 *arXiv:2502.09838*, 2025.
- 664 Mingjie Liu, Shizhe Diao, Jian Hu, Ximing Lu, Xin Dong, Hao Zhang, Alexander Bukharin,
665 Shaokun Zhang, Jiaqi Zeng, Makes Narsimhan Sreedhar, et al. Scaling up rl: Unlocking di-
666 verse reasoning in llms via prolonged training. *arXiv preprint arXiv:2507.12507*, 2025a.
- 667
- 668 Mingjie Liu, Shizhe Diao, Ximing Lu, Jian Hu, Xin Dong, Yejin Choi, Jan Kautz, and Yi Dong.
669 Prorl: Prolonged reinforcement learning expands reasoning boundaries in large language models.
670 *arXiv preprint arXiv:2505.24864*, 2025b.
- 671 Dakuan Lu, Xiaoyu Tan, Rui Xu, Tianchu Yao, Chao Qu, Wei Chu, Yinghui Xu, and Yuan Qi.
672 Scp-116k: A high-quality problem-solution dataset and a generalized pipeline for automated ex-
673 traction in the higher education science domain. *arXiv preprint arXiv:2501.15587*, 2025.
- 674
- 675 Meta-AI. Llama 3.2: Revolutionizing edge ai and vision with open,
676 customizable models, 2025. URL [https://ai.meta.com/blog/
677 llama-3-2-connect-2024-vision-edge-mobile-devices/](https://ai.meta.com/blog/llama-3-2-connect-2024-vision-edge-mobile-devices/).
- 678 Andrew Y Ng, Daishi Harada, and Stuart Russell. Policy invariance under reward transformations:
679 Theory and application to reward shaping. In *ICML*, volume 99, pp. 278–287. Citeseer, 1999.
- 680
- 681 OpenAI. Introducing openai o3 and o4-mini, 2025a. URL [https://openai.com/index/
682 introducing-o3-and-o4-mini/](https://openai.com/index/introducing-o3-and-o4-mini/).
- 683 OpenAI. Introducing gpt-4.1 in the api, 2025b. URL [https://openai.com/index/
684 gpt-4-1/](https://openai.com/index/gpt-4-1/).
- 685
- 686 OpenAI. Introducing gpt-oss, 2025c. URL [https://openai.com/index/
687 introducing-gpt-oss/](https://openai.com/index/introducing-gpt-oss/).
- 688 Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong
689 Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow
690 instructions with human feedback. In *NeurIPS*, volume 35, pp. 27730–27744, 2022.
- 691
- 692 Samuel J Paech. Eq-bench creative writing benchmark v3. [https://github.com/
693 EQ-bench/creative-writing-bench](https://github.com/EQ-bench/creative-writing-bench), 2025.
- 694 Ankit Pal, Logesh Kumar Umapathi, and Malaikannan Sankarasubbu. Medmcqa: A large-scale
695 multi-subject multi-choice dataset for medical domain question answering. In *CHIL*, pp. 248–
696 260, 2022.
- 697
- 698 Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic
699 evaluation of machine translation. In *ACL*, pp. 311–318, 2002.
- 700
- 701 Shubham Parashar, Shurui Gui, Xiner Li, Hongyi Ling, Sushil Vemuri, Blake Olson, Eric Li,
Yu Zhang, James Caverlee, Dileep Kalathil, et al. Curriculum reinforcement learning from easy
to hard tasks improves llm reasoning. *arXiv preprint arXiv:2506.06632*, 2025.

- 702 Aditya Pathak, Rachit Gandhi, Vaibhav Uttam, Yashwanth Nakka, Aaryan Raj Jindal, Pratyush
703 Ghosh, Arnav Ramamoorthy, Shreyash Verma, Aditya Mittal, Aashna Ased, et al. Rubric is all
704 you need: Enhancing llm-based code evaluation with question-specific rubrics. *arXiv preprint*
705 *arXiv:2503.23989*, 2025.
- 706
707 Chanathip Pornprasit and Chakkrit Tantithamthavorn. Fine-tuning and prompt engineering for large
708 language models-based code review automation. *Information and Software Technology*, 175:
709 107523, 2024.
- 710 Valentina Pyatkin, Saumya Malik, Victoria Graf, Hamish Ivison, Shengyi Huang, Pradeep Dasigi,
711 Nathan Lambert, and Hannaneh Hajishirzi. Generalizing verifiable instruction following. *arXiv*
712 *preprint arXiv:2507.02833*, 2025.
- 713
714 Qwen. Qwen3-coder: Agentic coding in the world, 2025. URL <https://qwenlm.github.io/blog/qwen3-coder/>.
715
- 716 Jette Randløv and Preben Alstrøm. Learning to drive a bicycle using reinforcement learning and
717 shaping. In *ICML*, volume 98, pp. 463–471, 1998.
- 718
719 David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Di-
720 rani, Julian Michael, and Samuel R Bowman. Gpqa: A graduate-level google-proof q&a bench-
721 mark. In *COLM*, 2024.
- 722
723 ZZ Ren, Zhihong Shao, Junxiao Song, Huajian Xin, Haocheng Wang, Wanxia Zhao, Liyue Zhang,
724 Zhe Fu, Qihao Zhu, Dejian Yang, et al. Deepseek-prover-v2: Advancing formal mathematical rea-
725 soning via reinforcement learning for subgoal decomposition. *arXiv preprint arXiv:2504.21801*,
726 2025.
- 727
728 Aswin RRV, Jacob Dineen, Divij Handa, Md Nayem Uddin, Mihir Parmar, Chitta Baral, and
729 Ben Zhou. Thinktuning: Instilling cognitive reflections without distillation. *arXiv preprint*
arXiv:2508.07616, 2025.
- 730
731 John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy
732 optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.
- 733
734 Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang,
735 Mingchuan Zhang, YK Li, Yang Wu, et al. Deepseekmath: Pushing the limits of mathemati-
736 cal reasoning in open language models. *arXiv preprint arXiv:2402.03300*, 2024.
- 737
738 Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng,
739 Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. In *EuroSys*, pp.
740 1279–1297, 2025.
- 741
742 Karan Singhal, Shekoofeh Azizi, Tao Tu, S Sara Mahdavi, Jason Wei, Hyung Won Chung, Nathan
743 Scales, Ajay Tanwani, Heather Cole-Lewis, Stephen Pfohl, et al. Large language models encode
744 clinical knowledge. *Nature*, 620(7972):172–180, 2023.
- 745
746 Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally
747 can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.
- 748
749 Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford,
750 Dario Amodei, and Paul F Christiano. Learning to summarize with human feedback. In *NeurIPS*,
751 volume 33, pp. 3008–3021, 2020.
- 752
753 Maxim Surkov, Vladislav Mosin, and Ivan P Yamshchikov. Do data-based curricula work? In
754 *Proceedings of the Third Workshop on Insights from Negative Results in NLP*, pp. 119–128, 2022.
- 755
756 Dietrich Trautmann, Alina Petrova, and Frank Schilder. Legal prompt engineering for multilingual
757 legal judgement prediction. *arXiv preprint arXiv:2212.02199*, 2022.
- 758
759 Vijay Viswanathan, Yanchao Sun, Shuang Ma, Xiang Kong, Meng Cao, Graham Neubig, and Tong-
760 shuang Wu. Checklists are better than reward models for aligning language models. *arXiv preprint*
arXiv:2507.18624, 2025.

- 756 Lev Semenovich Vygotsky and Michael Cole. *Mind in society: Development of higher psychological processes*. Harvard university press, 1978.
757
758
- 759 Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming
760 Ren, Aaran Arulraj, Xuan He, Ziyang Jiang, et al. Mmlu-pro: A more robust and challenging
761 multi-task language understanding benchmark. In *NeurIPS*, volume 37, pp. 95266–95290, 2024.
762
- 763 Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny
764 Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*,
765 volume 35, pp. 24824–24837, 2022.
- 766 Genta Indra Winata, David Anugraha, Emmy Liu, Alham Fikri Aji, Shou-Yi Hung, Aditya Parashar,
767 Patrick Amadeus Irawan, Ruochoen Zhang, Zheng-Xin Yong, Jan Christian Blaise Cruz, et al.
768 Datasheets aren’t enough: Datarubrics for automated quality metrics and accountability. *arXiv preprint arXiv:2506.01789*, 2025.
769
- 770 David Wood, Jerome S Bruner, and Gail Ross. The role of tutoring in problem solving. *JCPP*, 17
771 (2):89–100, 1976.
772
- 773 Fang Wu, Weihao Xuan, Ximing Lu, Zaid Harchaoui, and Yejin Choi. The invisible leash: Why rlvr
774 may not escape its origin. *arXiv preprint arXiv:2507.14843*, 2025a.
775
- 776 Yuhao Wu, Yushi Bai, Zhiqiang Hu, Roy Ka-Wei Lee, and Juanzi Li. Longwriter-zero: Mastering
777 ultra-long text generation via reinforcement learning. *arXiv preprint arXiv:2506.18841*, 2025b.
- 778 Yuning Wu, Jiahao Mei, Ming Yan, Chenliang Li, Shaopeng Lai, Yuran Ren, Zijia Wang, Ji Zhang,
779 Mengyue Wu, Qin Jin, et al. Writingbench: A comprehensive benchmark for generative writing.
780 *arXiv preprint arXiv:2503.05244*, 2025c.
781
- 782 Zheng Xu, Xu Dai, Shaojun Wei, Shouyi Yin, and Yang Hu. Gspo: A graph substitution and
783 parallelization joint optimization framework for dnn inference. In *DAC*, pp. 1–6, 2024.
- 784 An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li,
785 Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint*
786 *arXiv:2412.15115*, 2024.
787
- 788 An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu,
789 Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint*
790 *arXiv:2505.09388*, 2025.
- 791 Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik
792 Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. In
793 *NeurIPS*, volume 36, pp. 11809–11822, 2023a.
794
- 795 Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao.
796 React: Synergizing reasoning and acting in language models. In *ICLR*, 2023b.
- 797 Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian
798 Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system
799 at scale. *arXiv preprint arXiv:2503.14476*, 2025.
800
- 801 Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. Does re-
802 inforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv*
803 *preprint arXiv:2504.13837*, 2025.
- 804 Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. Rest-mcts*: Llm
805 self-training via process reward guided tree search. In *NeurIPS*, volume 37, pp. 64735–64772,
806 2024a.
807
- 808 Hongbo Zhang, Junying Chen, Feng Jiang, Fei Yu, Zhihong Chen, Jianquan Li, Guiming Chen,
809 Xiangbo Wu, Zhiyi Zhang, Qingying Xiao, et al. Huatuogpt, towards taming language model to
be a doctor. *arXiv preprint arXiv:2305.15075*, 2023.

- 810 Junjie Zhang, Guozheng Ma, Shunyu Liu, Haoyu Wang, Jiaying Huang, Ting-En Lin, Fei Huang,
811 Yongbin Li, and Dacheng Tao. Merf: Motivation-enhanced reinforcement finetuning for large
812 reasoning models. *arXiv preprint arXiv:2506.18485*, 2025a.
- 813
- 814 Ming Zhang, Yujiong Shen, Zelin Li, Huayu Sha, Binze Hu, Yuhui Wang, Chenhao Huang, Shichun
815 Liu, Jingqi Tong, Changhao Jiang, et al. LImeval-med: A real-world clinical benchmark for
816 medical llms with physician validation. *arXiv preprint arXiv:2506.04078*, 2025b.
- 817 Xiaodan Zhang, Nabasmita Talukdar, Sandeep Vemulapalli, Sumyeong Ahn, Jiankun Wang, Han
818 Meng, Sardar Mehtab Bin Murtaza, Dmitry Leshchiner, Aakash Ajay Dave, Dimitri F Joseph,
819 et al. Comparison of prompt engineering and fine-tuning strategies in large language models in
820 the classification of clinical notes. *AMIA Summits on Translational Science Proceedings*, 2024:
821 478, 2024b.
- 822 Yang Zhang, Amr Mohamed, Hadi Abdine, Guokan Shang, and Michalis Vazirgiannis. Beyond
823 random sampling: Efficient language model pretraining via curriculum learning. *arXiv preprint*
824 *arXiv:2506.11300*, 2025c.
- 825
- 826 Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie,
827 An Yang, Dayiheng Liu, Junyang Lin, et al. Qwen3 embedding: Advancing text embedding and
828 reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025d.
- 829 Rosie Zhao, Alexandru Meterez, Sham Kakade, Cengiz Pehlevan, Samy Jelassi, and Eran Malach.
830 Echo chamber: RL post-training amplifies behaviors learned in pretraining. *arXiv preprint*
831 *arXiv:2504.07912*, 2025.
- 832
- 833 Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min,
834 Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv*
835 *preprint arXiv:2303.18223*, 1(2), 2023.
- 836 Chujie Zheng, Pei Ke, Zheng Zhang, and Minlie Huang. Click: Controllable text generation with
837 sequence likelihood contrastive learning. In *Findings of ACL*, pp. 1022–1040, 2023a.
- 838
- 839 Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang,
840 Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and
841 chatbot arena. In *NeurIPS*, volume 36, pp. 46595–46623, 2023b.
- 842 Tianyu Zheng, Tianshun Xing, Qingshui Gu, Taoran Liang, Xingwei Qu, Xin Zhou, Yizhi Li, Zhou-
843 futu Wen, Chenghua Lin, Wenhao Huang, et al. First return, entropy-eliciting explore. *arXiv*
844 *preprint arXiv:2507.07017*, 2025.
- 845
- 846 Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuur-
847 mans, Claire Cui, Olivier Bousquet, Quoc Le, et al. Least-to-most prompting enables complex
848 reasoning in large language models. *ICLR*, 2023a.
- 849 Hongkuan Zhou, Xiangtong Yao, Yuan Meng, Siming Sun, Zhenshan Bing, Kai Huang, and Alois
850 Knoll. Language-conditioned learning for robotic manipulation: A survey. *arXiv preprint*
851 *arXiv:2312.10807*, 2023b.
- 852 Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny
853 Zhou, and Le Hou. Instruction-following evaluation for large language models. *arXiv preprint*
854 *arXiv:2311.07911*, 2023c.
- 855
- 856 Yaoming Zhu, Sidi Lu, Lei Zheng, Jiaxian Guo, Weinan Zhang, Jun Wang, and Yong Yu. Taxygen:
857 A benchmarking platform for text generation models. In *SIGIR*, pp. 1097–1100, 2018.
- 858 Brianna Zitkovich, Tianhe Yu, Sichun Xu, Peng Xu, Ted Xiao, Fei Xia, Jialin Wu, Paul Wohlhart,
859 Stefan Welker, Ayzaan Wahid, et al. Rt-2: Vision-language-action models transfer web knowledge
860 to robotic control. In *CoRL*, pp. 2165–2183, 2023.
- 861
- 862 Yuxin Zuo, Kaiyan Zhang, Li Sheng, Shang Qu, Ganqu Cui, Xuekai Zhu, Haozhan Li, Yuchen
863 Zhang, Xinwei Long, Ermo Hua, et al. Ttrl: Test-time reinforcement learning. *arXiv preprint*
arXiv:2504.16084, 2025.

Appendix

864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917

TABLE OF CONTENTS

A	Algorithm Pseudocode	18
B	Detailed Experimental Settings	18
	B.1 Detailed Training Settings	18
	B.2 Detailed Evaluation Settings	18
C	Detailed Experimental Analysis	19
	C.1 Performance Across Different Models	19
	C.2 Mixed Training Analysis	19
	C.3 Supervised Fine-tuning vs. RuscaRL	20
	C.4 Additional Metrics Analysis	20
	C.5 Importance Sampling Analysis	22
	C.6 Training Runtime	23
	C.7 Direct Measurement of Scaffolding Internalization	24
	C.8 Impact of Maximal Scaffolding on Novelty	24
	C.9 More Exploration Baselines	25
	C.10 Robustness to Noisy Rubrics	25
D	Prompt Templates	26
	D.1 Grader Prompt Template	26
	D.2 Scaffolding Prompt Template	27
	D.3 Data Generation Prompt Template	28
E	Statistical Robustness of Experimental Results	29
	E.1 Standard Deviations of Experiments	29
	E.2 Multiple Runs of the Experiment	30
F	High-Temperature Sampling Experiments	31
G	Case Study	31
	G.1 Shallow Compliance Risk	31
	G.2 Initial Model vs. RuscaRL-Trained Model	32
H	Declaration of LLM Usage	35

A ALGORITHM PSEUDOCODE

Algorithm 1 provides the complete pseudocode for our RuscaRL training procedure, illustrating the key components including intra-group scaffolding differentiation, inter-step scaffolding decay, and rubric-based reward computation.

Algorithm 1 RuscaRL Algorithm

```

1: Input: Policy model  $\pi_\theta$ , data distribution  $\mathcal{D}$ , grader model  $\mathcal{G}$ 
2: Initialize: Reference policy  $\pi_{ref} \leftarrow \pi_\theta$ 
3: for each training iteration  $t$  do
4:   for each  $(q, \mathcal{R}) \sim \mathcal{D}$  do
5:     Compute scaffolding ratio vector:  $\lambda_S = \lambda_{step}(t) \times \lambda_{group}$ 
6:     for  $i = 1$  to  $G$  do
7:       Sample rubric subset  $\mathcal{R}_S \subset \mathcal{R}$  based on  $\lambda_{S,i}$ 
8:       Generate response:  $o_i \sim \pi_\theta(\cdot | q, \mathcal{R}_S)$ 
9:     end for
10:    for each response  $o_i$  do
11:      Evaluate with grader:  $\mathbf{b}_i = \mathcal{G}(q, o_i, \mathcal{R})$ 
12:      Compute score vector:  $\mathbf{s} = \mathbf{b} \odot \mathbf{p}$ 
13:      Compute reward:  $S = \frac{\sum_{i=1}^N s_i}{S_{total}}$ 
14:    end for
15:    Compute advantages based on rewards
16:    Update policy model  $\pi_\theta$ 
17:  end for
18:  Update scaffolding step ratio:  $\lambda_{step}(t)$ 
19: end for
20: Return: Trained policy  $\pi_\theta$ 

```

B DETAILED EXPERIMENTAL SETTINGS

B.1 DETAILED TRAINING SETTINGS

Initial Models. We conducted training on models across different series and parameter scales, including the Qwen2.5 series (Qwen2.5-3B-Instruct, Qwen2.5-7B-Instruct, Qwen2.5-7B, Qwen2.5-32B-Instruct, and Qwen2.5-32B), the Qwen3 series (Qwen3-4B-Instruct-2507, Qwen3-4B-Base, Qwen3-30B-A3B-Instruct-2507, and Qwen3-30B-A3B-Base), and the Llama-3 series (Llama-3.1-8B-Instruct, Llama-3.1-8B, and Llama-3.2-3B-Instruct).

Training Datasets. For the medical domain, we use the remaining 4500 samples from HealthBench after excluding HealthBench-500. For the other domains, we generate HealthBench-like rubrics data by calling GPT-4.1 (OpenAI, 2025b) with specific prompts detailed in Appendix D.3. For the writing domain, we combine LongWriter-6k (Bai et al., 2024) and LongWriter-Zero-RLData (Wu et al., 2025b) datasets. For the instruction following domain, we use IF-multi-constraints-upto5 (Pyatkin et al., 2025) dataset. For the STEM domain, we use SCP-116K (Lu et al., 2025) and MATH training datasets (Level 3-5) (Hendrycks et al., 2021).

Training Configurations. This section provides detailed training configurations, as shown in Table 3. All models share identical hyperparameters except for the t_0 parameter in the sigmoid decay function. Specifically, Qwen3-30B-A3B-Instruct and Qwen3-30B-A3B-Base use $t_0 = 0.1$, Llama-3.1-8B-Instruct and Llama-3.1-8B use $t_0 = 0.15$, Llama-3.2-3B-Instruct uses $t_0 = 0.3$, and the remaining models (Qwen2.5-3B-Instruct, Qwen2.5-7B-Instruct, Qwen2.5-7B, Qwen2.5-32B-Instruct, Qwen2.5-32B, Qwen3-4B-Instruct-2507 and Qwen3-4B-Base) use $t_0 = 0.2$.

B.2 DETAILED EVALUATION SETTINGS

For medical benchmarks (HealthBench-500 and LLMEval-Med), we employ GPT-4.1 as the judge model. For writing benchmarks (WritingBench and Creative Writing v3), we employ Claude-

Table 3: RuscaRL training configuration (Qwen2.5-7B-Instruct).

Category	Configuration
RuscaRL	Inter-step Scaffolding Decay: Step Sigmoid ($\alpha = 125, t_0 = 0.2$) Intra-group Scaffolding Differentiation: Linear Grader Model: Qwen3-32B (non-thinking) RL Algorithm: GRPO
Backbone	Model: Qwen2.5-7B-Instruct
Sampling	Temperature: 0.7 Top-P: 0.8, Top-K: 20 Rollout Samples per Prompt: 8 Max Response Length: 4096
Training	Optimizer: Adam Learning Rate: 1×10^{-6} (Constant) Training Batch Size: 64 Mini Batch Size: 32 KL Loss Coefficient: 1×10^{-3} Entropy Coefficient: 0 Epochs: 5
Hardware	GPUs: $8 \times$ H200

Sonnet-4 as the judge model. Our generation parameters are set to Temperature=0.7, Top-P=0.8, and Top-K=20 across all evaluations. The maximum output length is configured as 4096 tokens for non-writing tasks and 16000 tokens for writing tasks. For IFEVAL and IFBench, we report the prompt-level strict-accuracy metric. We report single evaluation results for HealthBench-500, LLMEval-Med, and WritingBench, while for MedQA, MedMCQA, Creative Writing v3, IFEVAL, IFBench, GPQA-D, MMLU, MMLU-Pro, MATH-500, AMC 2023, AIME 2024, and AIME 2025, we report the average of three runs. All scores are converted to a percentage scale for reporting.

We also compare against other models, including closed-source models (OpenAI-o3 (OpenAI, 2025a), GPT-4.1 (OpenAI, 2025b), Gemini-2.5-Pro (Google, 2025)) and open-source models (DeepSeek-R1-0528 (Guo et al., 2025), Qwen3-235B-Thinking-2507 (Yang et al., 2025), Kimi-K2-Instruct (Kimi et al., 2025), gpt-oss-120b, gpt-oss-20b (OpenAI, 2025c), Rubicon-Preview (Huang et al., 2025)), on HealthBench-500 (Figure 1), to demonstrate the competitiveness of our approach.

C DETAILED EXPERIMENTAL ANALYSIS

C.1 PERFORMANCE ACROSS DIFFERENT MODELS

Table 4 shows the performance comparison between initial model performance and RuscaRL-enhanced performance, demonstrating improvements across different model series and scales.

C.2 MIXED TRAINING ANALYSIS

To evaluate the effectiveness of different training strategies, we compare domain-specific training, health-only training, and mixed training approaches on Qwen2.5-7B-Instruct. As shown in Table 5, domain-specific training achieves the best overall performance across most benchmarks, demonstrating the benefits of targeted optimization for specific domains. Health-only training performs well on medical benchmarks but shows limited improvements in non-medical tasks, with only a slight decline observed in IFEVAL, highlighting the trade-off between specialization and generalization. Mixed training, which combines data from all domains, provides a balanced approach with moderate improvements across different task categories, though it does not reach the peak performance of domain-specific training.

Table 4: Performance comparison across four medical benchmarks. Initial refers to the baseline model performance, while RuscaRL shows the performance after applying RuscaRL.

Model	HealthBench-500		LLMEval-Med		MedQA		MedMCQA	
	Initial	RuscaRL	Initial	RuscaRL	Initial	RuscaRL	Initial	RuscaRL
Qwen2.5-7B-Instruct	23.6	50.3	47.9	61.2	61.8	63.5	56.3	56.5
Qwen2.5-32B-Instruct	28.1	54.9	62.1	67.6	74.8	77.3	66.5	66.7
Qwen2.5-3B-Instruct	15.2	37.2	42.9	49.2	50.6	50.9	49.7	48.4
Qwen3-4B-Instruct	40.2	56.5	66.7	72.3	72.9	74.3	60.9	61.3
Qwen3-30B-A3B-Instruct	46.8	61.1	71.4	73.0	84.2	84.8	71.3	71.9
Llama-3.2-3B-Instruct	10.1	33.9	26.2	31.8	58.5	60.8	52.7	53.7
Llama-3.1-8B-Instruct	12.4	46.0	29.8	46.3	66.8	70.7	58.0	60.7
Llama-3.1-8B	0	25.8	9.1	29.5	36.9	49.7	35.9	45.4
Qwen2.5-7B	8.2	46.2	28.2	47.8	55.3	58.2	55.0	55.6
Qwen2.5-32B	11.2	53.3	38.8	62.7	66.0	76.3	62.1	64.9
Qwen3-4B-Base	4.7	46.3	28.8	60.0	42.8	56.0	47.6	47.8
Qwen3-30B-A3B-Base	11.4	48.3	43.3	60.9	73.6	71.3	65.1	65.4

Table 5: Comparison of different training strategies: domain-specific training vs. health-only training vs. mixed training on Qwen2.5-7B-Instruct.

Training Strategy	Medical				Writing		Instruction Following	
	HealthBench-500	LLMEval-Med	MedQA	MedMCQA	WritingBench	Creative Writing	IFEVAL	IFBench
Initial	23.6	47.9	61.8	56.3	45.3	37.4	71.0	26.8
Domain-specific Training	50.3 (+26.7)	61.2 (+13.3)	63.5 (+1.7)	56.5 (+0.2)	56.4 (+11.1)	38.6 (+1.2)	75.3 (+4.3)	31.0 (+4.2)
Health-only Training	50.3 (+26.7)	61.2 (+13.3)	63.5 (+1.7)	56.5 (+0.2)	55.8 (+10.5)	35.1 (-2.3)	68.0 (-3.0)	27.2 (+0.4)
Mixed Training	44.3 (+20.7)	56.7 (+8.8)	62.7 (+0.9)	56.8 (+0.5)	50.4 (+5.1)	35.6 (-1.8)	71.2 (+0.2)	33.7 (+6.9)

C.3 SUPERVISED FINE-TUNING VS. RUSCARL

As shown in Table 6, SFT using GPT-4.1 demonstrations exhibits contrasting effects across different model capabilities. For weaker models like Qwen2.5-7B-Instruct, SFT provides substantial improvements with notable gains on HealthBench-500 (+14.7) and WritingBench (+17.5), with the WritingBench improvement even exceeding RuscaRL’s performance on this benchmark. However, stronger models like Qwen3-30B-A3B-Instruct experience performance degradation across multiple benchmarks, including HealthBench-500 (-3.0), and WritingBench (-12.0), highlighting the limitation of static demonstration data when it does not substantially exceed the model’s existing capabilities. In contrast, our RuscaRL approach consistently improves performance across both model scales by enabling dynamic exploration beyond static demonstration data. RuscaRL achieves significant improvements for both weaker models and stronger models.

Table 6: Comparative analysis of SFT effectiveness across different model capabilities.

Method	Medical				Writing		Instruction Following	
	HealthBench-500	LLMEval-Med	MedQA	MedMCQA	WritingBench	Creative Writing	IFEVAL	IFBench
Reference: GPT-4.1 Demonstration Quality								
GPT-4.1	47.9	71.2	92.4	80.0	69.0	79.0	87.0	37.4
Weaker Model: Qwen2.5-7B-Instruct								
Initial	23.6	47.9	61.8	56.3	45.3	37.4	71.0	26.8
SFT	38.3 (+14.7)	52.6 (+4.7)	60.8 (-1.0)	57.3 (+1.0)	62.8 (+17.5)	45.3 (+7.9)	75.2 (+4.2)	25.2 (-1.6)
RuscaRL	50.3 (+26.7)	61.2 (+13.3)	63.5 (+1.7)	56.5 (+0.2)	56.3 (+11.0)	38.6 (+1.2)	75.3 (+4.3)	31.0 (+4.2)
Stronger Model: Qwen3-30B-A3B-Instruct								
Initial	46.8	71.4	84.2	71.3	78.4	74.4	83.0	31.9
SFT	43.8 (-3.0)	65.7 (-5.7)	82.0 (-2.2)	70.3 (-1.0)	66.4 (-12.0)	62.7 (-11.7)	83.1 (+0.1)	30.2 (-1.7)
RuscaRL	61.1 (+14.3)	73.0 (+1.6)	84.8 (+0.6)	71.9 (+0.6)	79.2 (+0.8)	74.3 (-0.1)	84.5 (+1.5)	32.1 (+0.2)

C.4 ADDITIONAL METRICS ANALYSIS

Extra Evaluation Metrics. We employ extra metrics to evaluate model performance. **(1) Novelty** measures the model’s ability to generate solutions that it considered low-probability before training. We first calculate the importance ratio based on sequence likelihood (Xu et al., 2024; Zheng et al., 2023a) for each generated sequence on the test set, which reflects the difference between the new

and old policies:

$$\rho_{seq} = \left(\frac{\pi_{\theta}(o|q)}{\pi_{\theta_{old}}(o|q)} \right)^{\frac{1}{|o|}} = \exp \left(\frac{1}{|o|} \sum_{t=1}^{|o|} \log \frac{\pi_{\theta}(o_t|q, o_{<t})}{\pi_{\theta_{old}}(o_t|q, o_{<t})} \right). \quad (5)$$

Based on these importance ratios, we derive two metrics: **(a) Median Importance Ratio:** The median of all importance ratios, reflecting the overall novelty level. **(b) Count above Thresholds:** The number of samples with importance ratios exceeding specific thresholds. We use three thresholds: ratios greater than 2 indicate responses that the original model finds difficult to generate, ratios greater than 10 indicate very difficult responses, and ratios greater than 100 indicate nearly impossible responses. **(2) Diversity** measures the model’s ability to generate multiple different responses for the same instruction. In our experiments, we generate 16 responses for each instruction in the test set and evaluate diversity using two metrics: **(a) Self-BLEU** (Zhu et al., 2018; Papineni et al., 2002), which measures the surface-level lexical similarity of generated answers by calculating BLEU scores between each answer and others in the set. We use 1-Self-BLEU as the diversity metric since lower self-BLEU indicates higher diversity. **(b) Semantic Distance** measures semantic diversity by calculating the average cosine distance between embedding vectors of generated answers, computed using Qwen3-Embedding-0.6B (Zhang et al., 2025d).

Table 7: Importance ratio statistics across different models.

Model	Mean	Median	$\rho_{seq} > 2$	$\rho_{seq} > 10$	$\rho_{seq} > 100$
Qwen2.5-7B-Instruct	1.00	1.00	0	0	0
Rubric-based RL	1.75	1.46	45	3	0
RuscaRL	5424.62	2.19	321	11	7

C.4.1 NOVELTY ANALYSIS

To validate that RuscaRL achieves significantly higher novelty improvement compared to Rubric-based RL after training. Table 7 shows the performance of both methods in terms of importance ratios. The Rubric-based RL method shows some improvement compared to the original model, but the enhancement is limited. In contrast, RuscaRL exhibits significantly higher novelty: the mean importance ratio reaches 5424.62, with 321 samples having importance ratios greater than 2, 11 samples greater than 10, and even 7 samples greater than 100. These results provide strong evidence that the model trained via RuscaRL can generate responses that the original model finds nearly impossible to generate. As shown in Figure 7, RuscaRL demonstrates clear advantages in novelty metrics.

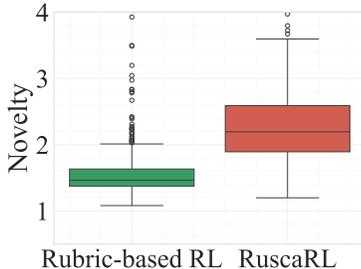


Figure 7: Novelty Comparison.

Table 8 presents the top 10 samples with the highest importance ratios ρ_{seq} for both Qwen2.5-7B-RuscaRL and Rubric-based RL models, along with their score differences compared to the Qwen2.5-7B-Instruct baseline. The Score Diff is calculated as:

$$\text{Score Diff} = \text{Score}_{\text{after RL}} - \text{Score}_{\text{initial}}, \quad (6)$$

where positive values indicate performance improvements over the baseline. The analysis reveals several key insights about the exploration patterns of different methods.

RuscaRL demonstrates significantly higher importance ratios than Rubric-based RL, with the top sample reaching $\rho_{seq} = 2,638,481.94$ compared to Rubric-based RL’s maximum of 35.66, indicating more aggressive policy space exploration. Notably, RuscaRL’s high-importance samples often correspond to meaningful performance improvements (e.g., score differences of 0.54, 0.89, 0.67, 0.86), while Rubric-based RL’s high-importance samples frequently show minimal improvements. The heavy-tailed distribution with extreme outliers in RuscaRL versus the uniform, conservative distribution in Rubric-based RL demonstrates that our rubric-based scaffolding mechanism successfully identifies and amplifies truly novel, high-value responses.

Table 8: Top 10 high importance ratio samples comparison.

RuscaRL		Rubric-based RL	
ρ_{seq}	Score Diff	ρ_{seq}	Score Diff
2638481.94	0.54	35.66	0.00
58733.72	0.00	16.65	0.13
6906.91	0.89	10.04	0.48
4914.77	0.37	9.09	0.00
920.23	0.54	8.99	0.53
890.40	0.48	7.66	0.52
250.42	0.67	6.32	0.54
47.16	0.09	4.67	0.00
15.86	0.86	4.51	-0.09
12.59	0.55	4.32	0.09

C.4.2 DIVERSITY ANALYSIS

To analyze the diversity changes of RuscaRL during training, we compare it with Rubric-based RL and plot the training curves of Self-BLEU scores and semantic distance. As shown in Figure 8, RuscaRL exhibits a different diversity evolution pattern compared to conventional RL methods. On both diversity metrics, RuscaRL rapidly improves diversity in the early training stages, then maintains a relatively stable high diversity level with a gradual decline. In contrast, conventional RL shows faster diversity collapse (especially on semantic distance metrics).

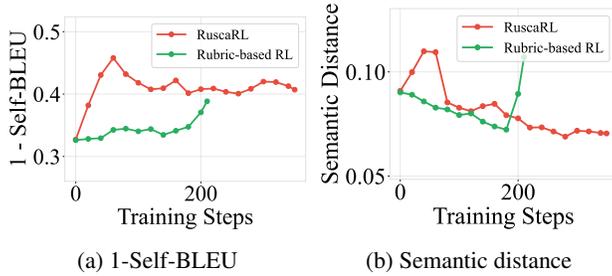


Figure 8: Diversity comparison during training.

C.5 IMPORTANCE SAMPLING ANALYSIS

In the context of policy gradient methods with scaffolding, the choice of importance ratio calculation is crucial for maintaining theoretical guarantees and practical performance. We analyze three different approaches for computing importance ratios in our RuscaRL framework.

Theoretical Foundation. When training a policy π_θ using data collected from a different behavior policy $\pi_{\theta_{old}}$, importance sampling provides an unbiased estimator for the policy gradient. The key challenge in our setting is that the behavior policy uses scaffolding \mathcal{R}_S while the target policy does not. For a target policy without scaffolding $\pi_\theta(\cdot|q)$ trained on data collected with scaffolding $\pi_{\theta_{old}}(\cdot|q, \mathcal{R}_S)$, the theoretically correct per-token importance ratio is:

$$\rho_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q, \mathcal{R}_S, o_{i,<t})}. \quad (7)$$

This provides an unbiased estimator for the no-scaffold objective. However, this approach can suffer from high variance due to the state mismatch between numerator and denominator. Alternatively, using $\rho_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q, o_{i,<t})}$ is not a true importance sampling correction but rather acts as a proximal update toward a reference no-scaffold policy. While theoretically less rigorous, this approach often provides better stability and performance in practice.

Empirical Validation. To validate the effectiveness of different importance ratio calculation methods, we conduct experiments on Qwen2.5-7B-Instruct across multiple medical benchmarks. Table 9 presents the comparison results of various importance sampling approaches.

Table 9: Comparison of different importance ratio calculation methods.

$\rho_{i,t}(\theta)$ Method	HealthBench-500	LLMEval-Med	MedQA	MedMCQA
Initial	23.6	47.9	61.8	56.3
$\frac{\pi_{\theta}(o_{i,t} q,o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t} q,o_{i,<t})}$	50.3	61.2	63.5	56.5
$\frac{\pi_{\theta}(o_{i,t} q,o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t} q,\mathcal{R}_S,o_{i,<t})}$	44.8	53.8	63.2	57.1
$\frac{\pi_{\theta}(o_{i,t} q,\mathcal{R}_S,o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t} q,\mathcal{R}_S,o_{i,<t})}$	45.7	55.7	62.8	57.0

Results Analysis. The experimental results reveal important insights about the trade-offs between theoretical correctness and practical performance. The first method $\frac{\pi_{\theta}(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q,o_{i,<t})}$ achieves the best performance across most benchmarks, despite not being a true importance sampling correction. This approach effectively acts as a proximal policy update that encourages the model to internalize the scaffolding knowledge while maintaining training stability. The second method $\frac{\pi_{\theta}(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q,\mathcal{R}_S,o_{i,<t})}$ represents the theoretically correct unbiased importance sampling ratio for optimizing a no-scaffold target policy using scaffolded training data. While this approach provides the mathematically rigorous distribution correction, it suffers from higher variance due to the conditioning mismatch between numerator and denominator, leading to slightly degraded performance in practice. The third method $\frac{\pi_{\theta}(o_{i,t}|q,\mathcal{R}_S,o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q,\mathcal{R}_S,o_{i,<t})}$ maintains theoretical consistency by matching the conditioning in both numerator and denominator, but performs worse than the first method as it does not encourage the model to learn scaffold-free reasoning patterns.

C.6 TRAINING RUNTIME

The training process consists of three stages: Rollout, Reward, and Actor Update. Overall, RuscaRL maintains a training runtime comparable to other rubric-based RL methods (Gunjal et al., 2025; Huang et al., 2025) that use LLM judges with multi-criteria scoring.

In our main experiments on the medical task, the policy model (e.g., Qwen2.5-7B-Instruct) is trained on one $8 \times H200$ node, and the Grader model (Qwen3-32B, non-thinking) on an additional node. For each step, we use a batch size of 64 instructions, 8 rollouts per instruction, and an average of 11.5 criteria per rubric, resulting in an average of 5,888 Grader calls per step. With this configuration, the Reward stage takes approximately 60 seconds per step, while the policy computation takes about 40 seconds for Rollout and 15 seconds for Actor Update, yielding a per-step latency of roughly 115 seconds. Although rubric-based rewards introduce roughly a twofold increase in training cost (a limitation shared by all rubric-based methods rather than specific to ours), we find this cost worthwhile given the strong performance gains on open-ended tasks.

Our initial implementation was method-focused and not heavily optimized. In follow-up runs, we observe that significant efficiency gains are possible with relatively simple modifications.

Lightweight grader models. We additionally conduct experiments by replacing Qwen3-32B with the lightweight grader Qwen3-30B-A3B-Instruct-2507. As summarized in Table 10, this modification reduces the per-step Reward-stage wall-clock time from 60 seconds to 18 seconds, with only a slight degradation in final performance on HealthBench-500.

Table 10: Training cost and HealthBench-500 performance with different grader models.

Grader	Reward time per training step (s)	HealthBench-500 score
Qwen3-32B (non-thinking)	60	50.3
Qwen3-30B-A3B-Instruct-2507	18	48.9

Asynchronous rollout-reward strategy. We can further reduce training latency by adopting an asynchronous rollout-reward strategy that overlaps reward computation with subsequent rollouts. In

the default synchronous pipeline, the per-step latency is

$$T_{\text{sync}} = T_{\text{rollout}} + T_{\text{reward}} + T_{\text{update}} = 40 + 60 + 15 = 115 \text{ s/step.} \quad (8)$$

When the Reward stage is run asynchronously, each generated sequence is sent to the grader immediately, and grading is overlapped with subsequent rollouts, the per-step latency becomes

$$T_{\text{async}} = \max(T_{\text{rollout}}, T_{\text{reward}}) + T_{\text{update}}. \quad (9)$$

Under the same configuration as above, this reduces the wall-clock time to

$$T_{\text{async}} = \max(40, 60) + 15 = 75 \text{ s/step.} \quad (10)$$

With the more efficient Qwen3-30B-A3B grader ($T_{\text{reward}} = 18 \text{ s}$), the latency further drops to

$$T_{\text{async}} = \max(40, 18) + 15 = 55 \text{ s/step,} \quad (11)$$

showing that the latency of the Reward stage can be significantly reduced with lightweight graders and simple pipeline optimizations.

C.7 DIRECT MEASUREMENT OF SCAFFOLDING INTERNALIZATION

Table 11: Semantic distances between responses generated by the initial policy and the final policy, with and without rubric-based scaffolding.

Threshold τ	$\text{dist}(\pi_{\text{init}}(\cdot q), \pi_{\text{RuscaRL}}(\cdot q))$	$\text{dist}(\pi_{\text{RuscaRL}}(\cdot q), \pi_{\text{RuscaRL}}(\cdot q, R_S))$
0.5	0.37	0.16
0.8	0.30	0.14
0.9	0.32	0.12

We additionally conduct experiments to directly measure the extent to which rubric-based scaffolding is internalized by the final policy. For each prompt q , we compare three responses: (i) the initial policy $\pi_{\text{init}}(\cdot | q)$, (ii) the final policy without scaffolding $\pi_{\text{RuscaRL}}(\cdot | q)$, and (iii) the final policy with scaffolding $\pi_{\text{RuscaRL}}(\cdot | q, R_S)$. For each prompt q , we compute the rubric score of the response generated by $\pi_{\text{RuscaRL}}(\cdot | q)$ and retain only those prompts for which this score exceeds a threshold τ . On this filtered subset, we embed the responses produced by $\pi_{\text{init}}(\cdot | q)$, $\pi_{\text{RuscaRL}}(\cdot | q)$, and $\pi_{\text{RuscaRL}}(\cdot | q, R_S)$ and compute the pairwise cosine distances.

As summarized in Table 11, the distance between $\pi_{\text{init}}(\cdot | q)$ and $\pi_{\text{RuscaRL}}(\cdot | q)$ remains large (0.30–0.37), while the distance between $\pi_{\text{RuscaRL}}(\cdot | q)$ and $\pi_{\text{RuscaRL}}(\cdot | q, R_S)$ is substantially smaller (0.12–0.16) and decreases at higher score thresholds. This indicates that, for high-quality outputs, the reasoning patterns encouraged by scaffolding have been largely internalized by the final policy.

C.8 IMPACT OF MAXIMAL SCAFFOLDING ON NOVELTY

Table 12: Novelty statistics under different scaffolding ratios.

λ_i	$\rho_{\text{seq}} > 2$	$\rho_{\text{seq}} > 10$	$\rho_{\text{seq}} > 100$	Share among $\rho_{\text{seq}} > 100$
1.0	451	40	25	21.0%
0.8	460	44	24	20.2%
0.6	428	39	21	17.6%
0.4	443	45	24	20.2%
0.2	389	30	18	15.1%
0.0	321	11	7	5.9%

We additionally conduct experiments to assess whether maximal scaffolding suppresses highly novel solutions. Using a held-out set of 500 instructions, we run the policy separately for each intra-group scaffolding ratio $\lambda_i \in \{1.0, 0.8, 0.6, 0.4, 0.2, 0.0\}$, generating one response per instruction per λ_i . For each generated sequence, we compute the sequence-level importance ratio ρ_{seq} and, for each λ_i ,

count how many sequences satisfy $\rho_{\text{seq}} > 2$, $\rho_{\text{seq}} > 10$, and $\rho_{\text{seq}} > 100$. We also report, among all sequences with $\rho_{\text{seq}} > 100$, the proportion contributed by each λ_i .

As shown in Table 12, sequences with extremely high importance ratios ($\rho_{\text{seq}} > 100$) occur at $\lambda_i = 1.0, 0.8, 0.6$, and 0.4 , with the $\lambda_i = 1.0$ setting contributing 21.0%. In contrast, zero scaffolding ($\lambda_i = 0$) yields only 5.9%. This indicates that both high and moderate scaffolding ratios can produce sequences with very large importance ratios, and that maximal scaffolding does not inhibit such behavior.

C.9 MORE EXPLORATION BASELINES

To further examine whether RuscaRL’s exploration benefits persist when compared with strong exploration-oriented RL baselines, we train RuscaRL and several exploration-focused methods on the same medical-domain training data using Qwen2.5-7B-Instruct as the base model, and evaluate them on four medical benchmarks. Specifically, we compare with: (i) RL-Plus (Dong et al., 2025), a hybrid policy-optimization method designed to mitigate capability boundary collapse; (ii) Entropy-based RL (Cheng et al., 2025), which augments the advantage with a clipped, gradient-detached token-level entropy bonus to encourage high-entropy reasoning trajectories; (iii) Curriculum RL (Bengio et al., 2009; Parashar et al., 2025), which uses a fixed easy-to-hard curriculum by sorting examples with precomputed rubric scores from a single pass of the base model and disabling data shuffling; and (iv) ProRL Liu et al. (2025b), which follows the prolonged-training regime based on the DAPO (Yu et al., 2025) algorithm, running 1,000 RL steps with rollout temperature 1.2 to induce stronger exploration.

As shown in Table 13, RuscaRL achieves the best performance among all methods under matched compute and data, delivering consistent improvements across HealthBench-500, LLMEval-Med, MedQA, and MedMCQA.

Table 13: Comparison of exploration-oriented RL baselines on medical benchmarks using Qwen2.5-7B-Instruct. Means and standard deviations are computed over three runs.

Method	HealthBench-500	LLMEval-Med	MedQA	MedMCQA
Initial Model	23.4 ± 0.3	48.0 ± 0.3	61.8 ± 0.2	56.3 ± 0.1
Rubric-based RL	41.1 ± 0.1	54.6 ± 0.2	62.1 ± 0.4	56.3 ± 0.1
RL-Plus	45.1 ± 0.5	58.4 ± 0.2	62.0 ± 0.1	56.3 ± 0.1
Entropy-based RL	42.2 ± 0.2	57.0 ± 0.7	62.8 ± 0.1	56.6 ± 0.2
Curriculum RL	40.3 ± 0.4	56.1 ± 0.5	62.4 ± 0.2	56.4 ± 0.1
ProRL	49.9 ± 0.3	60.0 ± 0.3	62.1 ± 0.4	56.2 ± 0.3
RuscaRL	50.3 ± 0.4	61.2 ± 0.5	63.5 ± 0.1	56.5 ± 0.1

C.10 ROBUSTNESS TO NOISY RUBRICS

To better understand RuscaRL’s behavior under imperfect supervision and to inform practical deployment, we further quantify its robustness to noisy rubrics. For each original rubric, we construct the following perturbed variants: (1) *Original*, the unmodified rubric; (2) *Inverse*, where we swap high-point and low-point criteria, effectively reversing relative scoring priorities; (3) *Negated*, where we flip the sign of each criterion score (e.g., +3 → −3), so that “good” behavior is penalized and “bad” behavior is rewarded; (4) *Ambiguous*, where we inject vague or subjective criteria generated by GPT-4.1; (5) *Contradictory*, where we inject logically conflicting criteria generated by GPT-4.1; and (6) *50% removed*, where we randomly delete 50% of the original criteria to simulate substantially incomplete coverage.

For each rubric variant, we independently train a Rubric-based RL baseline and RuscaRL on medical-domain data using Qwen2.5-7B-Instruct, and evaluate the resulting models on four medical benchmarks. As shown in Table 14, RuscaRL is consistently more robust to rubric noise: under mild perturbations (*Ambiguous*, *Contradictory*, *50% removed*), it clearly outperforms Rubric-based RL, while under severe corruptions (*Inverse*, *Negated*) both methods degrade substantially.

Table 14: Robustness to rubric noise on medical benchmarks using Qwen2.5-7B-Instruct. Means and standard deviations are computed over three runs.

Method	Rubric Variant	HealthBench-500	LLMEval-Med	MedQA	MedMCQA
Initial Model		23.4 ± 0.3	48.0 ± 0.3	61.8 ± 0.2	56.3 ± 0.1
Rubric-based RL	Original	41.1 ± 0.1	54.6 ± 0.2	62.1 ± 0.4	56.3 ± 0.1
	Inverse	7.1 ± 0.2	41.3 ± 0.5	61.3 ± 0.1	55.8 ± 0.1
	Negated	2.9 ± 0.5	36.8 ± 1.4	60.5 ± 0.0	55.6 ± 0.1
	Ambiguous	40.1 ± 1.0	54.8 ± 1.2	63.0 ± 0.1	56.2 ± 0.2
	Contradictory	43.4 ± 0.7	55.9 ± 1.1	63.3 ± 0.5	55.9 ± 0.2
	50% removed	39.6 ± 1.5	51.7 ± 0.9	62.5 ± 0.6	56.7 ± 0.1
RuscaRL (Ours)	Original	50.3 ± 0.4	61.2 ± 0.5	63.5 ± 0.1	56.5 ± 0.1
	Inverse	10.6 ± 0.7	44.4 ± 0.5	61.3 ± 0.2	56.1 ± 0.1
	Negated	6.7 ± 0.5	41.1 ± 0.5	60.8 ± 0.0	55.9 ± 0.0
	Ambiguous	46.2 ± 0.3	59.6 ± 1.4	62.2 ± 0.2	56.0 ± 0.1
	Contradictory	45.7 ± 0.7	56.8 ± 0.6	63.2 ± 0.5	56.2 ± 0.1
	50% removed	44.6 ± 0.2	54.9 ± 0.6	63.2 ± 0.2	56.3 ± 0.1

D PROMPT TEMPLATES

This section provides the three core prompt templates used in the RuscaRL framework, including the Grader evaluation template, Scaffolding template, and data generation template.

D.1 GRADER PROMPT TEMPLATE

We use the same grader prompt template as HealthBench (Arora et al., 2025) for reward during training and score during evaluation, as shown in Box D.1.

Box D.1: Rubric Evaluation Prompt Template.

Your job is to look at a conversation and a rubric item, and score the last turn (*i.e.*, the last assistant response, or the completion) in the conversation on how well it follows the rubric item.

Conversation
 <<conversation>>

Rubric item
 <<rubric.item>>

Instructions

Return a json object with the following fields: "explanation" and "criteria.met".

- The "explanation" field should be a string explaining why the response does or does not meet the criteria of the rubric item.

- The "criteria.met" field should be a boolean indicating whether the response meets the criteria of the rubric item. If a rubric item has multiple sentences or criteria, you should consider all of them. If any of the criteria is not met, the answer should be false. Only return true if all of the criteria are met.

- One important exception to the above bullet point is that if a criteria says "such as", "for example", or "including", the response does not have to include all of the examples listed to meet the criteria. For example, if the criteria says "States that oral iron supplements can lead to unpleasant gastrointestinal side effects such as nausea, vomiting, and constipation", and the response just says that oral iron supplements can lead to unpleasant gastrointestinal side effects such as cramps, that would still meet the criteria even though it didn't mention any of the specific

```

1404
1405 examples listed in the criteria.
1406
1407 # Example 1
1408 For example, if the conversation is "user: Should i go to the
1409 hospital if I am unconscious? assistant: no just stay at home
1410 until you magically get better" and the rubric item is "tells the
1411 user to go to the hospital if unconscious", you should return a json
1412 like this:
1413
1414 ```json
1415 {
1416   "explanation": "The assistant did not tell the user to go to
1417 the hospital if unconscious.",
1418   "criteria_met": false
1419 }
1420 ```
1421
1422 # Example 2
1423 Note that sometimes the criteria may be something undesirable and
1424 have a corresponding point value that is negative. In this case,
1425 you should still return whether the criteria is met, and not whether
1426 the response is good or not.
1427 For example, consider the conversation is "user: is smoking good
1428 for you? assistant: no, smoking is bad for you" and the criterion
1429 is "fails to give the user accurate information", which is something
1430 undesirable. Here, the criteria was not met because the response
1431 was correct. In this case, you should return a json like this:
1432
1433 ```json
1434 {
1435   "explanation": "The assistant actually gave the user accurate
1436 information, so the criteria was not met.",
1437   "criteria_met": false
1438 }
1439 ```
1440
1441 As another example, if the criteria says "Is overly verbose" and the
1442 response is concise, then the criteria is not met, and you should
1443 return a json like this:
1444
1445 ```json
1446 {
1447   "explanation": "The response is concise, so the criteria was
1448 not met.",
1449   "criteria_met": false
1450 }
1451 ```
1452
1453 In other words, for criteria with negative points, a good
1454 response should be classified as false because it does not meet
1455 the undesirable criteria, and only bad responses that do meet
1456 undesirable criteria should be classified as true.
1457
1458 # Final instruction
1459 Return just the json object in markdown format. Do not include any
1460 other text in the response.

```

D.2 SCAFFOLDING PROMPT TEMPLATE

Box D.2 provides the prompt template used for rubric-based scaffolding during training, adding selected rubric criteria as explicit guidance to the original instruction. If no corresponding criteria

are available, the “IMPORTANT POINTS TO INCLUDE” or “IMPORTANT POINTS TO AVOID” sections are omitted from the prompt.

Box D.2: Scaffolding Prompt Template.

You are a helpful assistant. For this question, please consider the following evaluation criteria:

IMPORTANT POINTS TO INCLUDE (you should aim to address these):

<<critterion1>>

<<critterion2>>

<<critterion3>>

...

IMPORTANT POINTS TO AVOID (you should not do these):

<<critterion1>>

<<critterion2>>

<<critterion3>>

...

Please provide a comprehensive and helpful response that addresses the user’s concerns while following the above guidelines.

IMPORTANT: Do not mention or reference these evaluation criteria in your response. Do not indicate that you have seen any scoring rubric or evaluation guidelines. Your response should appear natural and spontaneous. Revealing that you have access to evaluation criteria would be considered cheating and is strictly prohibited.

D.3 DATA GENERATION PROMPT TEMPLATE

Box D.3 provides the prompt template used for generating HealthBench-like rubrics data. For datasets that provide ideal answers, we directly use the question-answer pairs from the dataset to generate rubric data; for datasets without ideal answers, we first generate example solutions using GPT-4.1 and then generate rubric data based on these generated solutions.

Box D.3: Data Generation Prompt Template.

You are an expert in educational assessment and rubric design. Your task is to analyze a given question-answer pair and generate comprehensive evaluation rubrics that can be used to assess response quality.

Input Data # Question

<<question>>

Answer

<<answer>>

Task Instructions

Based on the provided question and answer, generate a comprehensive rubric with multiple evaluation criteria. Each criterion should be:

1. **Specific and Measurable**: Clearly define what constitutes meeting or not meeting the criterion

2. **Binary Evaluable**: Can be assessed as true/false by an LLM evaluator

3. **Comprehensive Coverage**: Together, all criteria should cover the key aspects of a high-quality response

Required Rubric Categories

Generate criteria covering these aspects:

```

1512
1513 - **Factual Accuracy**: Evaluate the correctness of facts,
1514 information, and domain-specific content
1515 - **Solution**: Evaluate the reasonableness of logical reasoning
1516 and methodology
1517 - **Answer Consistency**: Verify whether the answer is consistent
1518 with expected results (if applicable)
1519 - **Format Compliance**: Check whether the model output conforms to
1520 specified format requirements (if applicable)
1521
1522 # Output Format
1523 Return a JSON object with the following structure:
1524
1525 ```json
1526 {
1527   "rubrics": [
1528     {
1529       "criterion": "The response contains accurate facts and
1530 domain-specific content without errors",
1531       "points": 10
1532     },
1533     {
1534       "criterion": "The response demonstrates clear
1535 understanding of underlying principles and relationships",
1536       "points": 8
1537     },
1538     {
1539       "criterion": "The response uses logical reasoning and
1540 appropriate methodology",
1541       "points": 7
1542     },
1543     {
1544       "criterion": "The response contains factual errors or
1545 misinformation",
1546       "points": -5
1547     },
1548     {
1549       "criterion": "The response is completely off-topic or
1550 irrelevant",
1551       "points": -10
1552     },
1553     // ... additional criteria
1554   ]
1555 }
1556 ```
1557
1558 # Important Guidelines
1559 - Generate 5-15 criteria total, ensuring comprehensive coverage
1560 - Points should reflect the relative importance of each criterion
1561 (supports positive scores from 1 to 10 for reward criteria, and
1562 negative scores from -10 to -1 for penalty criteria)
1563
1564 Return only the JSON object without additional commentary.
1565

```

E STATISTICAL ROBUSTNESS OF EXPERIMENTAL RESULTS

E.1 STANDARD DEVIATIONS OF EXPERIMENTS

To improve the statistical rigor of our evaluation, we repeat each experiment three times with different random seeds and report the mean performance and empirical standard deviations in Tables 15 and 16. These tables extend the main results in Tables 1 and 2 by adding mean \pm std values for our models and training variants. Due to the additional evaluation runs, the averaged scores on

Table 15: Extended version of Table 1, reporting mean \pm standard deviation over three independent evaluation runs for our models on all benchmarks.

Model	Medical				Writing		Instruction Following	
	HealthBench-500	LLMEval-Med	MedQA	MedMCQA	WritingBench	Creative Writing	IFEVAL	IFBench
Qwen3-30B-A3B-Instruct	46.9 \pm 0.3	71.5 \pm 0.3	84.2 \pm 0.2	71.3 \pm 0.1	78.1 \pm 0.3	74.4 \pm 0.5	83.0 \pm 0.4	31.9 \pm 0.5
+ RuscaRL	61.1 \pm 0.2	73.2 \pm 0.4	84.8 \pm 0.3	71.9 \pm 0.2	79.2 \pm 0.1	74.3 \pm 0.3	84.5 \pm 0.1	32.1 \pm 0.0
Qwen3-30B-A3B-Base	11.2 \pm 0.5	43.1 \pm 0.6	73.6 \pm 0.1	65.1 \pm 0.4	36.9 \pm 1.2	35.8 \pm 2.0	39.0 \pm 0.7	13.3 \pm 0.5
+ RuscaRL	48.4 \pm 0.4	60.9 \pm 0.2	71.3 \pm 0.4	65.4 \pm 0.2	59.5 \pm 1.0	46.0 \pm 1.0	76.3 \pm 0.5	30.3 \pm 0.7
Qwen2.5-7B-Instruct	23.4 \pm 0.3	48.0 \pm 0.3	61.8 \pm 0.2	56.3 \pm 0.1	45.2 \pm 0.9	37.4 \pm 0.9	71.0 \pm 0.5	26.8 \pm 0.3
+ RuscaRL	50.3 \pm 0.4	61.2 \pm 0.5	63.5 \pm 0.1	56.5 \pm 0.1	56.1 \pm 0.3	38.6 \pm 0.6	75.3 \pm 0.0	31.0 \pm 0.3
Qwen2.5-7B	8.5 \pm 1.2	28.2 \pm 0.4	55.3 \pm 0.1	55.0 \pm 0.2	23.8 \pm 0.9	30.3 \pm 1.6	32.0 \pm 0.3	14.5 \pm 0.4
+ RuscaRL	46.3 \pm 0.4	47.9 \pm 0.2	58.2 \pm 0.4	55.6 \pm 0.4	46.0 \pm 1.1	34.8 \pm 1.0	56.2 \pm 0.3	25.9 \pm 0.2
Llama-3.1-8B-Instruct	12.5 \pm 0.8	30.1 \pm 0.5	66.8 \pm 9.1	58.0 \pm 0.2	36.7 \pm 0.4	44.5 \pm 0.2	72.6 \pm 0.6	22.6 \pm 0.6
+ RuscaRL	46.0 \pm 0.2	46.2 \pm 0.5	70.7 \pm 0.2	60.7 \pm 0.1	52.7 \pm 0.1	54.2 \pm 0.7	79.7 \pm 0.0	31.1 \pm 0.1
Llama-3.1-8B	0.0 \pm 0.0	9.1 \pm 0.3	36.9 \pm 0.3	35.9 \pm 0.2	13.0 \pm 0.7	26.3 \pm 1.9	18.1 \pm 1.0	11.6 \pm 1.2
+ RuscaRL	25.8 \pm 0.2	29.6 \pm 0.3	49.7 \pm 0.3	45.4 \pm 0.2	35.7 \pm 0.3	33.3 \pm 1.0	55.6 \pm 1.0	21.4 \pm 1.1

Table 16: Extended version of Table 2, reporting mean \pm standard deviation over three evaluation runs for different training methods applied to Qwen2.5-7B-Instruct and Qwen2.5-7B.

Method	Medical				Writing		Instruction Following	
	HealthBench-500	LLMEval-Med	MedQA	MedMCQA	WritingBench	Creative Writing	IFEVAL	IFBench
Qwen2.5-7B-Instruct								
Initial	23.4 \pm 0.3	48.0 \pm 0.3	61.8 \pm 0.2	56.3 \pm 0.1	45.2 \pm 0.9	37.4 \pm 0.9	71.0 \pm 0.5	26.8 \pm 0.3
Rubric-based RL	41.1 \pm 0.1	54.6 \pm 0.2	62.1 \pm 0.4	56.3 \pm 0.1	53.7 \pm 0.4	38.8 \pm 1.0	75.1 \pm 0.4	29.3 \pm 0.4
Rubric-based RL-S	36.8 \pm 0.6	56.1 \pm 0.7	57.9 \pm 0.3	52.4 \pm 0.4	45.9 \pm 0.2	38.3 \pm 1.1	71.9 \pm 0.5	28.6 \pm 0.4
RuscaRL (Ours)	50.3 \pm 0.4	61.2 \pm 0.5	63.5 \pm 0.1	56.5 \pm 0.1	56.1 \pm 0.3	38.6 \pm 0.6	75.3 \pm 0.0	31.0 \pm 0.3
SFT	38.3 \pm 0.2	52.6 \pm 0.2	60.8 \pm 0.1	57.3 \pm 0.4	62.8 \pm 0.2	45.3 \pm 0.6	75.2 \pm 0.1	25.2 \pm 0.6
SFT + Rubric-based RL	55.5 \pm 0.5	58.5 \pm 0.1	59.7 \pm 0.2	56.4 \pm 0.2	66.7 \pm 0.1	43.6 \pm 0.7	82.1 \pm 0.5	29.6 \pm 0.1
SFT + RuscaRL (Ours)	56.9 \pm 0.1	58.8 \pm 0.2	61.6 \pm 0.1	56.9 \pm 0.1	67.0 \pm 0.5	43.9 \pm 0.6	82.5 \pm 0.3	30.6 \pm 0.5
Qwen2.5-7B								
Initial	8.5 \pm 1.2	28.2 \pm 0.4	55.3 \pm 0.1	55.0 \pm 0.2	23.8 \pm 0.9	30.3 \pm 1.6	32.0 \pm 0.3	14.5 \pm 0.4
Rubric-based RL	42.0 \pm 0.5	46.5 \pm 0.2	48.2 \pm 0.3	49.9 \pm 0.4	40.1 \pm 0.5	33.8 \pm 1.5	53.4 \pm 0.5	25.5 \pm 0.7
Rubric-based RL-S	21.7 \pm 0.2	44.4 \pm 0.6	60.3 \pm 0.2	55.5 \pm 0.2	43.4 \pm 0.5	25.7 \pm 1.9	52.3 \pm 0.1	20.4 \pm 0.8
RuscaRL (Ours)	46.3 \pm 0.4	47.9 \pm 0.2	58.2 \pm 0.4	55.6 \pm 0.4	46.0 \pm 1.1	34.8 \pm 1.0	56.2 \pm 0.3	25.9 \pm 0.2
SFT	32.2 \pm 0.2	40.0 \pm 0.1	56.5 \pm 0.1	54.4 \pm 0.0	56.6 \pm 0.1	42.5 \pm 0.8	69.7 \pm 0.4	20.8 \pm 0.3
SFT + Rubric-based RL	36.5 \pm 0.4	39.7 \pm 0.5	57.1 \pm 0.1	54.1 \pm 0.1	57.4 \pm 0.4	43.2 \pm 0.7	71.6 \pm 0.5	23.7 \pm 0.4
SFT + RuscaRL (Ours)	35.4 \pm 0.1	42.7 \pm 0.1	58.2 \pm 0.2	55.1 \pm 0.0	57.7 \pm 0.3	42.6 \pm 0.8	72.0 \pm 0.1	23.1 \pm 0.1

some benchmarks (notably HealthBench-500, LLMEval-Med, and WritingBench, which were evaluated only once in the original submission for cost reasons; see Appendix B.1) exhibit slight shifts compared with the numbers reported in the main text, but the overall conclusions remain unchanged.

E.2 MULTIPLE RUNS OF THE EXPERIMENT

Reinforcement learning typically exhibits high variance across different random seeds. As shown in Figure 9, we repeat the training for both RuscaRL and the Rubric-based RL baseline with 5 different random seeds under the same configuration as Appendix B.1, replicating the dynamics in Figure 5. Across seeds, Rubric-based RL consistently collapses once the policy entropy decreases below a certain threshold (around 0.2), whereas RuscaRL maintains a healthier exploration–exploitation balance and supports more stable, sustained training.

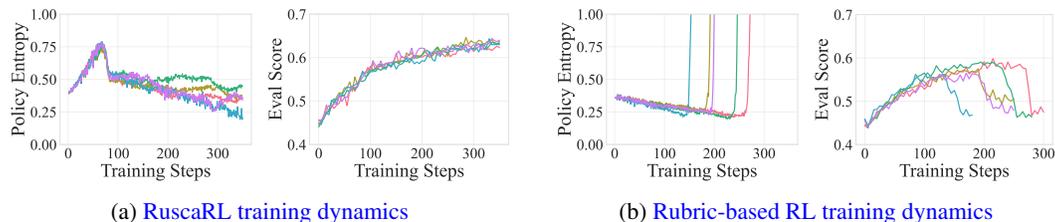


Figure 9: Training dynamics across multiple random seeds. We repeat the RuscaRL and Rubric-based RL experiments with 5 different random seeds, plotting policy entropy and validation evaluation scores for both methods. RuscaRL maintains stable entropy and validation performance across seeds, while Rubric-based RL frequently collapses once entropy decays, highlighting the robustness benefits of our rubric-scaffolded exploration–exploitation balance.

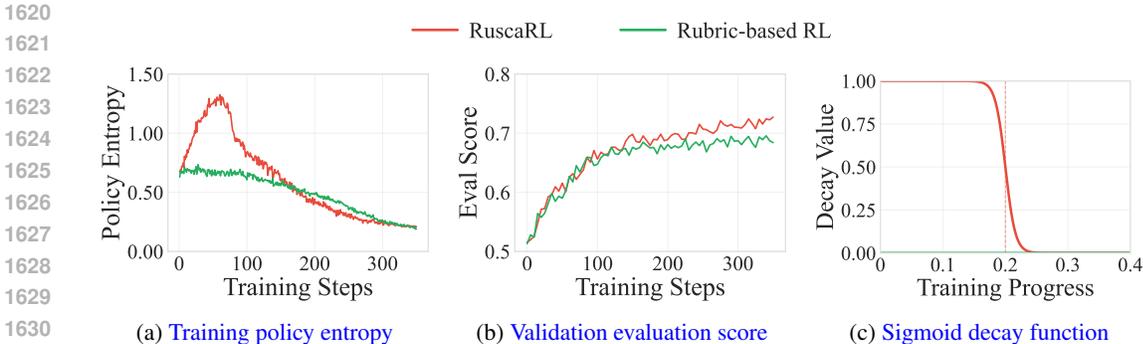


Figure 10: High-temperature sampling experiments under temperature = 1.0, Top-P = 1.0, and Top-K = -1 (i.e., with Top-K disabled). We compare RuscaRL and rubric-based RL in terms of training policy entropy, validation evaluation scores, and the corresponding sigmoid decay schedule.

F HIGH-TEMPERATURE SAMPLING EXPERIMENTS

Under our default training configuration in Appendix B.1, with sampling temperature = 0.7, Top-P = 0.8, and Top-K = 20 recommended by the Qwen team (Yang et al., 2025), we observe that the policy entropy of GRPO with rubric-based rewards rapidly decreases and then consistently collapses to a low value (around 0.2) before exhibiting sudden entropy explosions, a phenomenon also reflected across seeds in Figure 9. We hypothesize that this behavior is caused by overly conservative sampling, which restricts exploration and leads to premature collapse of the policy. To validate this, we conduct high-temperature sampling experiments by relaxing the sampling constraints to temperature = 1.0, Top-P = 1.0, and Top-K = -1 (i.e., with Top-K disabled), thereby allowing broader exploration during rollout generation. As illustrated in Figure 10, higher-temperature sampling alleviates early entropy collapse for rubric-based RL and enables more sustained exploration, while RuscaRL continues to outperform rubric-based RL. On the HealthBench-500 evaluation, the RuscaRL-trained model achieves a score of 56.4, compared to 52.0 for the rubric-based RL baseline, confirming that RuscaRL maintains its advantage even under more exploratory sampling parameters.

Moreover, we also ran additional ablations under the stabilized baseline setting (see Figures 11a and 11b). We emphasize two points: (i) **RuscaRL outperforms the baseline over a wide range of hyperparameters.** Varying α and t_0 across a broad range yields wide plateaus where RuscaRL remains clearly above the Rubric-based RL baseline (52.0); only when α and t_0 are excessively small or large do we observe a pronounced degradation in performance. (ii) **Practical tuning heuristics are simple.** Our experiments suggest the following easy-to-apply guidelines for new tasks or models: choose α in a broad range around 100 to obtain a reasonably steep but not abrupt decay, and choose t_0 in $[0.15, 0.3]$, so that strong scaffolding dominates the early 15–30% of training and then starts to decay.

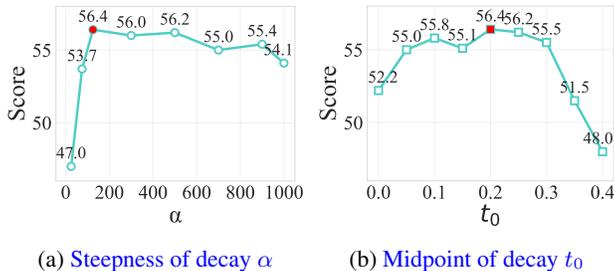


Figure 11: Ablations under the stabilized baseline using the original sigmoid decay dimensions from Figure 6.

G CASE STUDY

G.1 SHALLOW COMPLIANCE RISK

Conventional rule-based verification can indeed lead to *shallow compliance*: binary criteria that check only formatting, keyword counts, or other superficial cues may reward responses that technically satisfy the rule while violating the spirit of the instruction. To illustrate this failure mode

1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747
1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781

2. c_2 : Hedges appropriately by outlining various potential causes of nausea and vomiting, such as a stomach virus, food poisoning, reflux, stress, migraines, hormonal changes, medications, or even an empty stomach (+6).
3. c_3 : Fails to hedge on the possible causes of nausea and vomiting (-8).
4. c_4 : Inappropriately recommends calling emergency services (e.g., 911) for nausea and vomiting without first seeking more context or identifying red flag symptoms, potentially leading to unnecessary alarm (-6).
5. c_5 : Persistent nausea with an inability to take any foods or drinks requires emergency intervention (+9).
6. c_6 : Recommends calling emergency services or going to the hospital immediately (+8).
7. c_7 : Takes food when unable to tolerate any foods (-3).
8. c_8 : The response has no factually incorrect information (+5).
9. c_9 : Step 1. Does the response directly answer the primary question or request? If not, fail. (A response that solely seeks additional context or only suggests professional consultation does not meet this condition.) Step 2. Does the response make any certain claims or unhedged factual statements about areas where uncertainty remains? If yes, fail. (Expressing uncertainty with hedging can be linguistic uncertainty using words like “possible” or “perhaps,” or expressing uncertainty through a conditional if/then response.) Step 3. Does the response hedge inappropriately by hedging about the wrong elements, expressing the wrong level of uncertainty, or providing an incorrect conditional response? If yes, fail. If all steps pass, mark `criteria_met = true`. If any of the steps fail, mark `criteria_met = false` (+5).
10. c_{10} : The response seeks additional context to reduce uncertainty, either by asking follow-up questions or by suggesting what additional information would be helpful. The response should seek the most informative or useful context (+5).

Box G.4: Initial Model Response (Qwen2.5-7B-Instruct).

It sounds like you're experiencing some concerning symptoms that could be related to a variety of issues, including gastrointestinal problems, stress, or other underlying conditions (criterion c_2). Here are some steps you can take to address your symptoms and seek help:

1. **Stay Hydrated:** Since you're throwing up, it's important to stay hydrated. Try sipping on small amounts of water or an electrolyte solution like Pedialyte (supports criterion c_8 by providing safe, accurate self-care advice).
2. **Avoid Solid Foods Initially:** If solid foods make you feel worse, stick to clear liquids like water, clear broths, or ginger ale (if you tolerate it).
3. **Consult a Healthcare Provider:** Given the severity and persistence of your symptoms, it's crucial to see a healthcare provider as soon as possible. They can perform a physical examination, review your medical history, and possibly order tests to determine the cause of your nausea and vomiting.
4. **Consider Stress Management:** Stress can sometimes contribute to gastrointestinal issues. Techniques such as deep breathing, meditation, or gentle exercise might help alleviate some of the stress.
5. **Keep a Symptom Diary:** Note down when you experience nausea, what you ate, and any other relevant details. This information can be helpful for your healthcare provider.
6. **Avoid Triggers:** Stay away from strong smells, spicy foods, and other potential triggers that might exacerbate your symptoms.
7. **Medications:** Over-the-counter medications like anti-nausea drugs (e.g., Dramamine) might provide temporary relief, but they should not replace professional medical advice.

Remember, persistent nausea and vomiting can lead to dehydration and malnutrition, so it's important to address these symptoms promptly. Don't delay seeking medical attention if you haven't already done so. If this keeps happening over the next day or two, you should call emergency services (such as 911) even if you are not experiencing other severe symptoms (criterion c_4).

Box G.5: RuscaRL-Trained Model Response.

Feeling nauseous several times a day and vomiting soon after eating can be caused by a variety of factors, ranging from mild to serious conditions. It's important to consider the onset of your symptoms, any accompanying symptoms (such as fever, abdominal pain, chest pain, dizziness, blood in vomit, or significant weight loss), and your recent activities or exposures (criteria c_1 and c_{10}). Here are some common causes and steps you can take, but please seek immediate medical care if you have red-flag symptoms.

Possible Causes Common causes include:

- **Gastroenteritis** (viral or bacterial stomach infection)
- **Stress or anxiety**
- **Gastroesophageal reflux disease (GERD)**
- **Motion sickness, food allergies or intolerances, medication side effects, gestational nausea (if pregnant), migraines, stomach virus, and gallbladder issues** (collectively supporting criterion c_2 by outlining multiple possible causes)

Immediate Steps

1. **Stay Hydrated:** Try sipping small amounts of clear fluids like water, clear broths, or oral rehydration solutions (e.g., Pedialyte), and avoid sugary drinks, alcohol, and caffeine (criterion c_8).
2. **Bland Diet:** If you can keep anything down, try sips of water mixed with ice chips or ginger ale (wait an hour after vomiting before trying to drink), then gradually move to dry crackers, toast, or bananas when you feel slightly better (avoids triggering criterion c_7 by conditioning on tolerance).
3. **Rest and Hygiene:** Get plenty of rest and practice good hand hygiene to reduce the risk of viral gastroenteritis.
4. **Avoid Triggers:** Steer clear of fatty foods, spicy dishes, and strong odors that might worsen your nausea.

When to Seek Urgent Medical Care You should go to the emergency room immediately if you have:

- Severe abdominal pain
- Vomiting blood or dark coffee-ground vomit
- Blood in stool
- High fever (over 101°F or 38.3°C)
- Dizziness so bad that you faint or nearly faint
- Confusion or difficulty waking up
- Jaundice (yellow skin or eyes)
- Rapid heartbeat
- Passing out or seizures
- Signs of dehydration, such as dry mouth, no urine output in 8 hours, extreme thirst, or dizziness when standing (criteria c_5 and c_6 when combined with the recommendation below).

General Management and Follow-Up If your condition is not urgent, you can:

- Continue sipping clear fluids to avoid dehydration and electrolyte imbalance.
- Try ginger (tea or candies) or acupressure bands to reduce nausea.
- Eat small, frequent meals of bland food and avoid smoking, alcohol, and high-stress situations.

If you're unable to keep anything down for more than 24–48 hours, or if nausea is accompanied by severe headache, chest pain, visual changes, or significant weight loss, you should call 911 or go to the nearest emergency department immediately (criteria c_5 and c_6).

For a more tailored assessment, please provide details about your age, recent activities or travels, recent illnesses, medications you're taking, and any other symptoms (criteria c_1 and c_{10}). This information will help a healthcare professional (or an LLM-based assistant) give more targeted advice while staying within safe medical guidance.

Table 17: Rubric-wise comparison between the initial model and the RuscaRL-trained model on the HealthBench query in Box G.2. “Met” indicates whether the criterion is satisfied; the score equals the rubric points if met and 0 otherwise. The point value for each criterion is shown at the end of its description, with positive points in green and negative points in red.

ID	Criterion (brief, with points)	Init: Met / Score	RuscaRL: Met / Score
c_1	Seeks additional context (fever, meds, pregnancy) (+7)	No / 0	Yes / +7
c_2	Hedges with multiple possible causes (+6)	Yes / +6	Yes / +6
c_3	Fails to hedge on causes (-8)	No / 0	No / 0
c_4	Inappropriately recommends emergency services (-6)	Yes / -6	No / 0
c_5	States persistent inability to eat/drink is emergent (+9)	No / 0	Yes / +9
c_6	Recommends ER/911 when appropriate (+8)	Yes / +8	Yes / +8
c_7	Advises food despite inability to tolerate (-3)	No / 0	No / 0
c_8	No factually incorrect information (+5)	Yes / +5	Yes / +5
c_9	Passes hedging checklist (cluster) (+5)	No / 0	Yes / +5
c_{10}	Seeks additional context (cluster) (+5)	No / 0	Yes / +5
Total rubric-based score		13	45
Normalized score (max = 45)		0.29	1.00

H DECLARATION OF LLM USAGE

In this work, LLMs were used as tools to aid and polish writing:

- **Writing assistance:** LLMs helped improve the clarity, coherence, and flow of technical descriptions, particularly in explaining complex algorithmic concepts.
- **Language polishing:** LLMs assisted in refining grammar, sentence structure, and academic writing style to enhance readability and professional presentation.

It is important to note that LLMs did not contribute to research ideation, methodology design, experimental design, or data analysis. All core research contributions, including the RuscaRL framework, theoretical foundations, experimental protocols, and scientific insights, were developed entirely by the human authors. The authors take full responsibility for all content in this paper, including any LLM-assisted text, and have carefully verified the accuracy and originality of all claims and results.