RiT: Rubrics-in-Thinking Reinforcement Learning for Improved Reasoning in Large Language Models

Xiaobin Tian* **Shuai Yuan*** † **Muyun Ding** **Haonan Chen** **Xiaoxi Jiang**
Qwen Large Model Application Team, Alibaba
{tianxiaobin.txb, qianchun.ys, dingmuyun.dmy}@alibaba-inc.com
{haolan.chn, xiaoxi.jiang}@alibaba-inc.com

Abstract

Large Reasoning Models (LRMs) benefit from generating intermediate reasoning steps, enabling more reliable and interpretable decision-making. While outcome-based supervision has proven effective for LRMs across diverse tasks, it focuses solely on final answers and cannot guarantee high-quality intermediate reasoning. In contrast, existing process supervision is largely limited to verifiable domains such as mathematics or code, where intermediate steps can be explicitly checked, restricting its applicability to open-ended reasoning tasks. To address these limitations, we propose Rubrics-in-Thinking Reinforcement Learning (RiT), the first framework to introduce *thinking-rubric supervision* into intermediate reasoning. RiT automatically generates fine-grained rubrics and integrates them into a reward function via gated fusion with outcome-based rewards, guiding models to reason in a coherent and task-aligned manner, improving both intermediate steps and the final response. Experiments on reasoning-intensive and open-ended benchmarks demonstrate that RiT consistently outperforms outcome-only RL baselines.

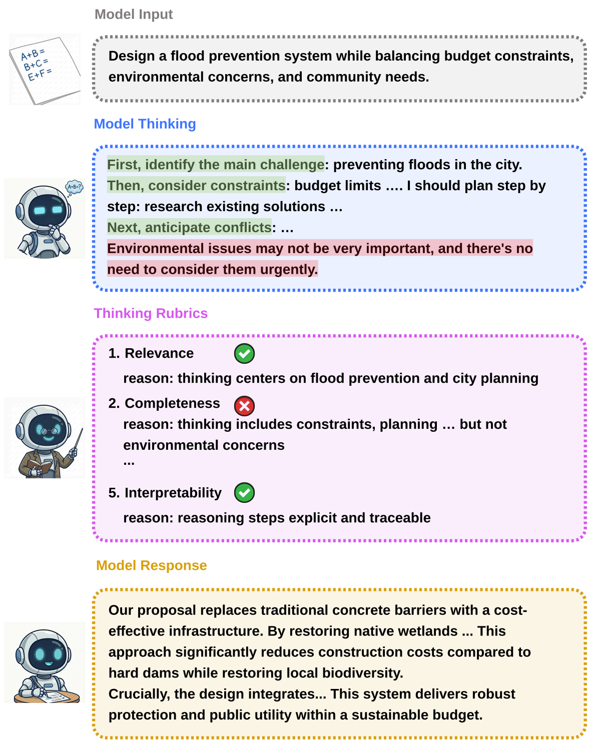

Figure 1: Thinking rubrics guide intermediate reasoning to maintain quality and task alignment.

1 Introduction

Large reasoning models (LRMs) (DeepSeek-AI, 2025; Jaech et al., 2024; Wei et al., 2022) have recently attracted attention for generating intermediate reasoning steps alongside final answers, enabling more reliable and interpretable decision-making. This capability benefits diverse natural language tasks, including mathematical problem solving (Mai et al., 2025), multi-step question answering (Lee and Hockenmaier, 2025), logical inference (Chen et al., 2025), and code generation (El-Kishky et al., 2025). However, most training paradigms for LRMs rely heavily on outcome-based rewards, optimizing primarily for final answer correctness. This can encourage shortcut reasoning, break multi-step logical consistency, and overlook qualitative aspects such as coherence and decisiveness. Process reward models (PRMs) (Khalifa et al., 2025) offer one solution but are often domain-specific (Lu et al., 2025; Fan et al., 2025; Yu et al., 2025b), relying on handcrafted heuristics or verifiable intermediate steps, and thus struggle to generalize to open-ended tasks involving subjective judgment or semantic interpretation. These limitations highlight a key challenge: **how can we guide a model's reasoning in a flexible yet principled way?**

One promising direction comes from rubric-based reinforcement learning (RL), which has recently shown success in guiding model behavior

[*] Equal contribution.
[†] Corresponding author.

Findings of the Association for Computational Linguistics: ACL 2026, pages 3944–3957
July 2-7, 2026 ©2026 Association for Computational Linguistics

through structured criteria for evaluating final responses (Gunjal et al., 2025; Liu et al., 2025; Huang et al., 2025). However, the application of rubrics to intermediate reasoning remains largely unexplored. As Figure 1 illustrates, extending rubric evaluation to the thinking process enables interpretable, multi-dimensional feedback on reasoning trajectories, improving both quality and reliability. Motivated by this, we propose Rubrics-in-Thinking RL (RiT), a simple and general framework that automatically generates fine-grained thinking rubrics and integrates them into a reward function for RL. This function evaluates reasoning trajectories according to rubrics and is fused with outcome-based rewards to balance the quality of reasoning and final response, encouraging more coherent and interpretable reasoning across tasks.

Our main contributions are:

- We systematically study thinking-rubric supervision in LRMs, analyzing its effects.
- We propose RiT, which automatically generates fine-grained rubrics and leverages them to supervise intermediate reasoning via RL.
- Experiments demonstrate the effectiveness and generalization of thinking-rubric supervision across diverse tasks and models.

2 Related Work

Large reasoning models (LRMs) (DeepSeek-AI, 2025; Marjanovic et al., 2025; Jaech et al., 2024) are a class of large language models (LLMs) designed to produce intermediate reasoning steps alongside final responses. By explicitly exposing reasoning chains, LRMs enable verification, planning, and reasoning-aware optimization, distinguishing them from conventional LLMs that output only end results (Chen et al., 2025; Xu et al., 2025). Existing LRMs have been primarily developed for narrow domains such as mathematical problem solving, code generation, and formal logic, often leveraging reward models or reinforcement learning to supervise the coherence and correctness of reasoning steps (Mai et al., 2025; El-Kishky et al., 2025; Fan et al., 2025; Lee and Hockenmaier, 2025; Khalifa et al., 2025; Yu et al., 2025b). However, these approaches rarely generalize to heterogeneous tasks, where inputs, outputs, and evaluation metrics vary significantly across domains, such as open-ended generation and general reasoning benchmarks. To address this limitation, we propose a thinking-rubric-based supervision framework that guides LRM training across diverse tasks, promoting reasoning-aware behavior beyond specialized domains.

Rubric-based RL has emerged as a paradigm for aligning LLMs by replacing scalar rewards with structured, multi-dimensional evaluation criteria. While traditional RL with Verifiable Rewards (RLVR) (Shao et al., 2024; Yu et al., 2025a; Zheng et al., 2025) excels in objective domains like math and code, recent works (Gunjal et al., 2025; Huang et al., 2025) show that rubrics can extend RL to open-ended reasoning tasks without ground truth. To reduce manual design, methods such as OpenRubrics (Liu et al., 2025) enable automated, scalable rubric generation, capturing nuanced human preferences. Rubrics further aid exploration and instruction following (Zhou et al., 2025), and have been specialized for high-stakes or subjective domains (Wang et al., 2025b), collectively advancing interpretable and robust reward modeling in RL for LLM. Notably, however, no prior work has systematically investigated the specific impact of thinking rubrics on model reasoning performance.

Open-ended generation tasks (Que et al., 2024; Bai et al., 2025; Lin et al., 2025) such as long-form and professional writing have become a primary focus of recent research as model capabilities advance beyond traditional closed-form settings. These tasks lack a single canonical target and instead require satisfying multiple qualitative criteria. Traditional metrics are often inadequate (Liu et al., 2023b), motivating the adoption of LLM-as-a-judge frameworks to assess generation quality (Gu et al., 2024; Bai et al., 2024). Methodologically, prior work on open-ended generation primarily adopts either data-centric enhancement or outcome-oriented reinforcement learning (Pham et al., 2024; Jia et al., 2025; Wang et al., 2025a).

3 Preliminaries

Formalization of LRMs. Given an input query q, an LRM generates an output sequence that consists of an explicit reasoning component and a final answer. Formally, the output is defined as:

$$y = \langle r, o \rangle, \tag{1}$$

where r denotes a sequence of intermediate reasoning steps and o denotes the final response. r is explicitly delimited by a special structural token

</think>, which separates the internal reasoning process from the final answer. In this work, we refer to r as the *thinking process* of the LRM.

Reinforcement Learning for LRMs. We define $\pi_\theta(y \mid q)$ as a parameterized policy mapping an input query q to an output sequence y. Reinforcement learning is employed to optimize the policy with respect to a scalar reward function $R(q, y)$, which evaluates the quality of the generated output, including both the final response and the associated reasoning process. Specifically, the learning objective is defined as:

$$\max_\theta \mathbb{E}_{y \sim \pi_\theta(\cdot|q)} \left[R(q, y) \right]. \tag{2}$$

4 Methodology

4.1 Overview

In this section, we present our proposed RiT framework. Our key insight is that effective reasoning in LRMs necessitates the explicit regulation of intermediate thinking processes, rather than merely optimizing the final response quality. To this end, we propose a unified training framework that integrates thinking-aware reward modeling with group-relative optimization, enabling stable and interpretable learning under heterogeneous reward signals. The technical implementation of RiT consists of three key components, as illustrated in Figure 2. First, we construct a set of **Thinking Rubrics** that specify task-dependent criteria for evaluating thinking processes, explicitly defining what constitutes desirable reasoning behaviors under different queries. Second, based on these rubrics, we design a **Gated Fusion Reward** that reflects both thinking and response quality while mitigating spurious high rewards from superficial patterns. Finally, we optimize the model using **Group Relative Policy Optimization (GRPO)** (Shao et al., 2024), where the fused reward serves as the training signal and group-wise normalization enables stable learning across heterogeneous reward scales.

4.2 Thinking Rubrics Generation

The first step in RiT is to generate **thinking rubrics**, i.e., explicit criteria that guide and evaluate the model's intermediate reasoning. Reasoning in LRMs is often under-specified or unstructured, which can result in incomplete, inconsistent, or logically flawed trajectories. By providing these rubrics, we establish clear standards for high-quality reasoning, enabling reliable evaluation of intermediate outputs and effectively guiding subsequent RL. To generate the rubrics, a frontier LLM is prompted with instructions comprising:

- **Task Definition.** The prompt specifies the task requirements and, when applicable, the evaluation dimensions of the final response (e.g., factual correctness, relevance, completeness). This ensures that the generated rubrics are aligned with the task objectives and provide explicit guidance for reasoning.

- **Positive Examples.** Model-generated reasoning trajectories that successfully satisfy the task requirements. These exemplify desirable reasoning behaviors and facilitate the LLM in inferring high-quality rubrics.

- **Negative Examples.** Reasoning trajectories that fail to meet the task requirements. These illustrate common errors, enabling the LLM to generate rubrics that effectively distinguish correct from incorrect reasoning.

After generating an initial rubric set, we further employ the LLM to iteratively refine the rubrics through a self-evaluation process. Specifically, the LLM is instructed to examine whether the current rubrics suffer from *missing criteria*, *redundant or overlapping conditions*, or *ambiguous*. Based on this analysis, the LLM revises the rubrics by adding missing aspects, removing or merging suboptimal criteria, and improving clarity and coverage.

The resulting thinking rubrics are denoted as:

$$\mathcal{C}_{\text{think}} = \{c_1, c_2, \ldots, c_K\}, \tag{3}$$

where each c_k represents an explicit criterion that defines a desirable aspect of reasoning (e.g., logical coherence, relevance, completeness). These rubrics form the foundation for computing the thinking reward $R_{\text{thinking}}(r)$, ensuring that the model's reasoning trajectories are structured, task-aligned, and interpretable. Importantly, this design is flexible and general, applicable to a wide range of tasks including open-ended generation and structured reasoning without task-specific modifications.

4.3 Reward Design

Thinking Rubric Reward. This reward quantifies the quality of the model's intermediate reasoning, explicitly incentivizing trajectories that

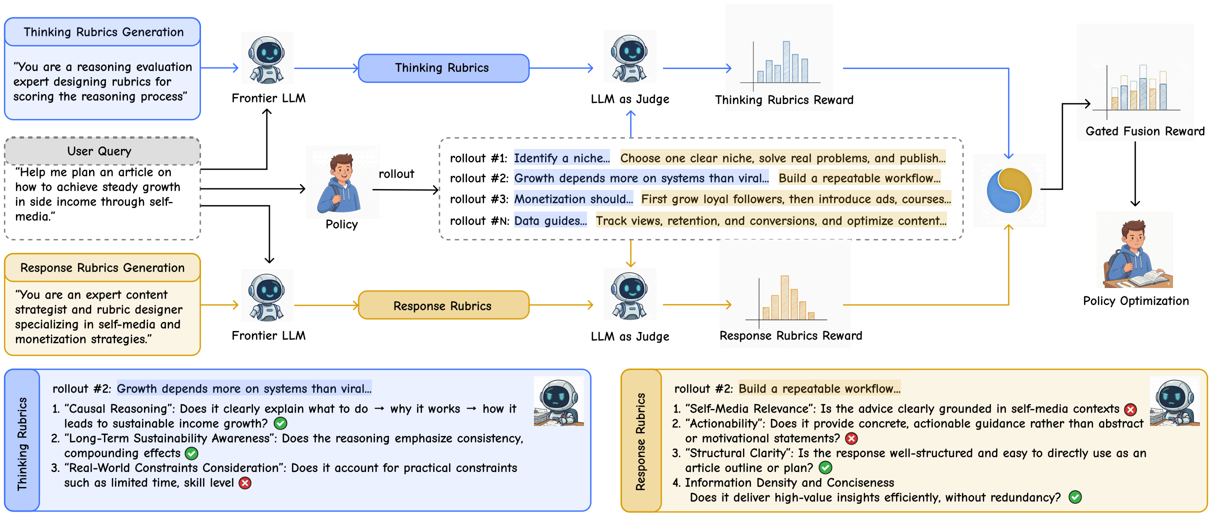

Figure 2: Overview of RiT, where a thinking-rubric reward is integrated with the response reward via gated fusion.

are well-formed, coherent, and aligned with task-specific criteria. High-quality reasoning not only improves the final response but also enhances interpretability and controllability of model behavior.

For a reasoning trajectory r corresponding to query q, its performance is evaluated against the rubrics set $\mathcal{C}_{\text{think}}$. Formally, the thinking rubric reward is defined as:

$$R_{\text{thinking}}(r) = \frac{1}{K}\sum_{k=1}^{K} s_k^{think}, \tag{4}$$

where $s_k^{think} \in \{0,1\}$ indicates whether r satisfies the k-th criterion under the LLM-based evaluator. By leveraging a frontier LLM to score the rubrics, this design ensures evaluations that are both robust and adaptive, capturing nuanced aspects of reasoning quality that are challenging to specify manually. Grounding the reward function in these structured rubrics allows RiT to guide the model toward producing trajectories that are simultaneously high-quality and interpretable, establishing a principled foundation for subsequent RL.

Response Reward. This reward evaluates the quality of the model's final response and adapts to different task types:

- **Open-ended generation tasks:** For tasks without a unique correct answer, we employ **response rubrics**, analogous to the thinking rubrics, to assess multiple dimensions of the output. These rubrics are constructed according to the evaluation criteria of the underlying dataset, ensuring alignment with task-specific standards. A frontier LLM critic assigns binary scores for each rubric, and the aggregated response reward is computed as:

$$R_{\text{resp}} = \frac{1}{M}\sum_{m=1}^{M} s_m^{\text{resp}}, \tag{5}$$

where $s_m^{\text{resp}} \in \{0,1\}$ denotes whether the response meets the m-th criterion of M rubrics.

- **Reasoning tasks:** For tasks with a well-defined answer, the response reward is based on **accuracy**, i.e., whether the model's output matches the reference answer. Formally,

$$R_{\text{resp}} = \mathbf{1}\{o_{\text{model}} = o_{\text{gold}}\}, \tag{6}$$

where o_{model} and o_{gold} denote the model-generated and gold answer, respectively.

Gated Fusion Reward. To jointly optimize both the reasoning process and the final response, we integrate the thinking reward R_{thinking} and the response reward R_{resp} through a weighted fusion inspired by (Yu et al., 2025b; Peng et al., 2025), followed by a gated mechanism that enforces balanced performance. Specifically, let $\alpha \in [0,1]$ denote the weight assigned to the reasoning component, with $1-\alpha$ corresponding to the response component (see Appendix A.4 for details on α):

$$R_{\text{fusion}} = \alpha R_{\text{thinking}} + (1-\alpha) R_{\text{resp}}. \tag{7}$$

The fused score is then passed through a **gating function** to obtain the final reward:

$$R_{\text{final}} = \text{Gate}(R_{\text{fusion}}, R_{\text{resp}}). \tag{8}$$

In this work, we instantiate the gating function as a hard minimum, such that the final reward is bounded by the weaker of the two components. The gating mechanism prevents degenerate optimization by adaptively constraining the influence of reasoning signals. This design ensures that high reward can only be achieved when both the reasoning process and the final response exhibit strong performance. Consequently, the model is discouraged from over-optimizing either reasoning or output in isolation, while retaining flexibility for future extensions with soft or adaptive gating mechanisms.

4.4 Reinforcement Learning with GRPO

To optimize the proposed reward formulation, we adopt GRPO as the RL algorithm. Given a query q, the policy π_θ samples a group of candidate responses $\{y_i\}_{i=1}^{g}$. Each response is assigned a scalar reward $R(q, y_i)$ computed using the proposed gated fusion of thinking and response scores. GRPO then calculates a group-relative advantage by normalizing rewards within each group:

$$A_i = \frac{R(q, y_i) - \bar{R}_g}{\mathrm{std}_{j \in g}(R(q, y_j))}, \quad (9)$$

where $\bar{R}_g = \frac{1}{|g|} \sum_{i \in g} R(q, y_i)$ and std denotes the standard deviation over the group. Based on the normalized advantage A_i, the policy is optimized by maximizing a clipped surrogate objective:

$$\mathcal{L}_{\text{GRPO}}(\theta) = \mathbb{E}_i \Big[\min \big(\rho_i(\theta) A_i,\ \mathrm{clip}(\rho_i(\theta), 1 - \epsilon, 1 + \epsilon) A_i \big) \Big], \quad (10)$$

where $\rho_i(\theta) = \frac{\pi_\theta(y_i|q)}{\pi_{\theta_{\text{old}}}(y_i|q)}$, ϵ is a hyperparameter controlling the clipping range to prevent excessively large policy updates.

5 Experiments

In this section, we conduct a detailed evaluation of our method RiT. Implementation details are provided in the Appendix A.1 and the source code is available at link[1].

5.1 Experimental Setup

Datasets. We evaluate RiT on diverse datasets spanning open-ended generation and reasoning tasks. For open-ended generation, we use **LongBench-Write** (Bai et al., 2025), which targets long-form text requiring coherent and contextually rich outputs, and **WritingBench** (Wu et al., 2025), a multi-domain dataset covering six professional domains, including technical, creative, and business writing. For reasoning, we employ **StrategyQA** (Geva et al., 2021), consisting of multi-step strategic questions; **LogiQA** (Liu et al., 2023a), designed for logical deduction and inference; and **TruthfulQA** (Lin et al., 2022), which assesses the truthfulness of model-generated answers. RiT is trained on the available training sets and evaluated on the corresponding test sets. Dataset statistics and details are provided in the Appendix A.2.

Backbone Reasoning Models. We conduct experiments using three backbone reasoning models: **Qwen3-8B** (Yang et al., 2025), **Qwen3-1.7B** (Yang et al., 2025), and **DeepSeek-R1-Distill-Llama-8B** (DeepSeek-AI, 2025). These models cover different parameter scales and architectures, providing a representative set for evaluating the effectiveness and generality of our thinking rubric supervision.

Baselines. To rigorously assess the effectiveness of RiT, we consider two representative baselines. First, we evaluate the **base model**, which serves as a performance lower bound. Second, we construct a **GRPO baseline with outcome-only reward (ORM)**, where optimization depends exclusively on response quality and provides no explicit signal on the thinking process. This baseline shares the same optimization algorithm as our method but differs only in reward design. Importantly, our goal is not to achieve state-of-the-art performance, but to validate the utility of the thinking rubric reward.

Evaluation Metrics. For reasoning tasks, we use the standard Pass@1 (Chen et al., 2021) metric. For open-ended generation, we employ Qwen3-235B-A22B-Instruct (Yang et al., 2025) as the judge for all benchmarks. More details of evaluation are provided in the Appendix A.3.

5.2 Main Results

Table 1 reports the main experimental results across different reasoning models and benchmark settings. Overall, RiT consistently outperforms all baselines across diverse evaluation scenarios, demonstrating strong generalization. In open-ended generation settings, it exhibits improved handling of long-form composition, producing responses that better satisfy complex writing conditions. In reasoning-oriented tasks, it achieves higher accuracy and stronger factual reasoning capability, reflecting more reliable and consistent decision-making be-

[1] https://github.com/Qwen-Applications/RiT

Reasoning Models	Methods	Open-Ended Generation		Reasoning		
		LongBench-Write	WritingBench	StrategyQA	LogiQA	TruthfulQA
DeepSeek-R1-Distilled -Llama-8B	Base	54.9	4.07	0.563	0.463	0.504
	ORM	59.5	4.46	0.659	0.558	0.579
	RiT	**60.3**	**4.70**	**0.703**	**0.564**	**0.695**
Qwen3-1.7B	Base	74.5	6.37	0.638	0.571	0.500
	ORM	76.3	7.10	0.716	0.597	0.848
	RiT	**79.3**	**7.13**	**0.734**	**0.606**	**0.854**
Qwen3-8B	Base	84.4	7.45	0.729	0.594	0.744
	ORM	85.7	8.10	0.808	0.690	0.945
	RiT	**89.2**	**8.24**	**0.838**	**0.719**	**0.951**

Table 1: Comparative evaluation of different methods across reasoning models and datasets. Best ones among all methods are **bold-faced**, and second-best are underlined.

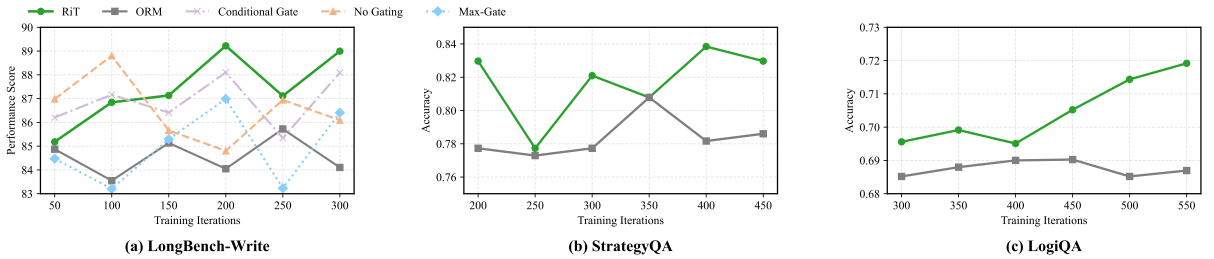

Figure 3: (a) LongBench-Write: ablation on both the thinking-rubric reward and the choice of gating function. (b) StrategyQA and (c) LogiQA: ablation on the thinking-rubric reward.

havior. Importantly, these performance gains are observed across models with varying parameter scales and architectural designs, ranging from lightweight models to larger, more expressive architectures. This suggests that RiT is not tightly coupled to a specific model size or backbone, and can be effectively applied across different model families, highlighting its robustness and scalability. Notably, ORM can be viewed as an ablation of RiT without the thinking-rubric reward. Comparing this variant with RiT underscores the contribution of the thinking-rubric guidance, demonstrating that incorporating reasoning-quality feedback significantly improves performance across diverse tasks. We further observe that models with stronger base capabilities benefit more from the thinking-rubric guidance. This is likely because high-capacity models can better leverage the fine-grained feedback to produce more coherent, accurate, and complex outputs, whereas lower-capacity models may struggle to fully exploit the reward signal.

5.3 Ablation Study

Module Analysis. Table 2 presents an ablation study on Qwen3-8B across three benchmarks: LongBench-Write, StrategyQA, and LogiQA. Our analysis reveals four main insights: (1) The full RiT consistently outperforms all ablation variants, demonstrating that TR (Thinking-Rubric Reward) and RR (Response Reward) are not redundant but mutually reinforcing. (2) RR acts as the primary driver of performance in reasoning tasks, highlighting that direct feedback on the final answer is essential for ensuring logical correctness and correcting terminal errors. (3) For open-ended tasks that lack a unique "gold" label, TR provides crucial guidance, effectively serving as a "compass" for intermediate reasoning. (4) Replacing GRPO with standard PPO (Schulman et al., 2017) results in a performance drop, suggesting that GRPO is better suited to balancing the heterogeneous reward scales introduced by the combination of thinking and response-level supervisions. (5) We further replace the thinking-rubric reward with a PRM-style step-level reward (Khalifa et al., 2025; Xiong et al., 2025), while keeping all training settings and the judge model identical to RiT for a fair comparison. The resulting variant still falls short of RiT, indicating that the gains of RiT stem from structured, task-aligned rubric guidance rather than step-level

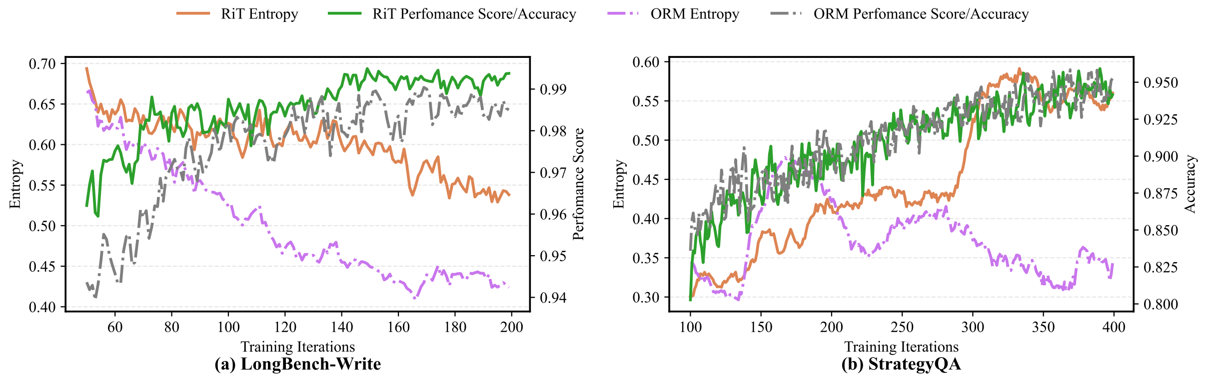

Figure 4: Entropy and performance of Qwen3-8B with RiT and ORM on LongBench-Write (a) and StrategyQA (b).

supervision or reward fusion alone.

Fine-grained Ablation of the Thinking-Rubric Reward. To further analyze the effect of the thinking-rubric reward, we evaluate Qwen3-8B on three representative benchmarks, including LongBench-Write, StrategyQA, and LogiQA. We compare RiT with ORM using checkpoints near reward convergence, enabling a controlled examination of their performance trajectories. As observed in the Figure 3, during the late convergence stages, RiT maintains performance scores that are generally not lower than ORM, with relatively minor fluctuations. This indicates that RiT demonstrates relatively stable superiority during the convergence phase, suggesting both high performance and more reliable behavior as training stabilizes.

Impact of Gate Function Choice. We further analyze the impact of different gating strategies through ablation experiments on Qwen3-8B evaluated on the LongBench-Write, as illustrated in Figure 3 (a). Specifically, we compare several alternatives: (1) **No Gating**, where the final reward is simply the weighted fusion of thinking rubric and response scores; (2) a **Max-Gate**, which selects the larger of the two scores; and (3) a **Conditional Gate**, where the thinking-rubric reward is incorporated only when the response is fully correct (i.e., the response score equals 1). Empirically, we observe that most gated variants outperform the outcome-only baseline, indicating that incorporating thinking rubric signals generally benefits learning. However, among all designs, the Min-Gate (RiT) consistently yields the most stable and reliable improvements. In contrast, the Max-Gate often leads to degenerate behaviors, as high thinking rubric scores can dominate even when the final answer is incorrect or low quality, weakening coupling between thinking quality and task success.

Methods	LongBench-Write	StrategyQA	LogiQA
Base	84.4	0.729	0.594
RiT (w/o TR)	85.7	0.808	0.690
RiT (w/o RR)	85.8	0.769	0.638
RiT (PPO)	<u>88.0</u>	0.799	0.638
RiT (PRM)	86.5	<u>0.812</u>	<u>0.695</u>
RiT	**89.2**	**0.838**	**0.719**

Table 2: Ablation on Qwen3-8B across LongBench-Write, StrategyQA, and LogiQA. TR and RR denote the Thinking-Rubric and Response rewards, respectively. Best results are in **bold**, and second-best are <u>underlined</u>.

Impact of Rubric Quality across Generation Strategies and Models. To systematically evaluate the role of rubric quality, we vary both the rubric generation strategies and models, and examine their impact on downstream performance using Qwen3-8B as the base model. Figure 5 shows that rubric quality is both controllable and important: (1) it can be improved through iterative refinement, and (2) more capable rubric generation models produce higher-quality evaluation criteria for assessing reasoning, leading to consistent gains in downstream performance. These results suggest that the quality of rubrics directly affects the effectiveness of the reward signal used for optimization.

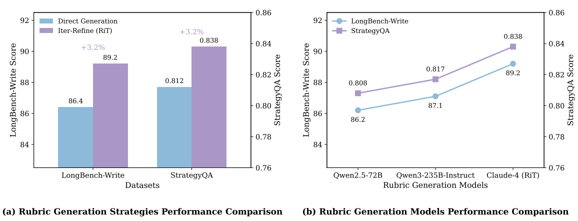

Figure 5: Impact of rubric quality on Qwen3-8B performance across the LongBench-Write and StrategyQA under rubric generation strategies (a) and models (b).

Model	StrategyQA			LogiQA			LongBench-Write		
	pass@1	pass@8	avg@8	pass@1	pass@8	avg@8	pass@1	max@8	avg@8
Base	0.729	0.873	0.743±0.016	0.594	0.746	0.611±0.003	84.43	93.59	86.45±1.71
ORM	0.808	0.886	0.787±0.017	0.690	0.796	0.689±0.004	85.70	91.20	84.96±1.07
RiT	**0.838**	**0.900**	**0.814±0.010**	**0.719**	**0.815**	**0.716±0.004**	**89.20**	**94.59**	**88.29±1.39**

Table 3: Results under multi-sample evaluation on reasoning and open-ended generation tasks. Values after ± denote standard deviations for avg@8 across repeated sampling runs.

5.4 Entropy Analysis

We further analyze the entropy dynamics of Qwen3-8B on the LongBench-Write and StrategyQA datasets, with a particular focus on the near-convergence stage of training. As illustrated in Figure 4, on LongBench-Write, the entropy of both methods exhibits an overall decreasing trend as training progresses. Nevertheless, RiT demonstrates a noticeably slower rate of entropy reduction compared to ORM, indicating its ability to preserve a higher degree of exploration during optimization. In contrast, on the reasoning-oriented StrategyQA task, RiT maintains an increasing entropy trend even near convergence, whereas ORM exhibits a clear decline in entropy. This suggests that RiT is able to sustain higher uncertainty and flexibility in its reasoning process, which may facilitate the exploration of diverse solution paths. Overall, these observations suggest that incorporating the thinking rubric endows the model with stronger exploratory capacity, enabling more effective reasoning behaviors across different task types. This enhanced ability to explore and deliberate allows the model to achieve improved performance in both open-ended generation and reasoning benchmarks.

5.5 Thinking Quality Comparison

We evaluate the thinking quality of Qwen3-8B under three strategies: Base, ORM, and RiT on LongBench-Write and StrategyQA. Following prior work, a frontier LLM (i.e., Claude Sonnet 4 (Anthropic, 2025)) selects the best reasoning trajectory for each query, and the selection ratio is used as the thinking quality metric. As shown in Figure 6, RiT consistently achieves the highest selection rate across both datasets, indicating more coherent and goal-aligned reasoning. This trend aligns with our entropy analysis, suggesting that thinking-guided training enhances exploratory capacity and overall reasoning quality beyond outcome-only supervision.

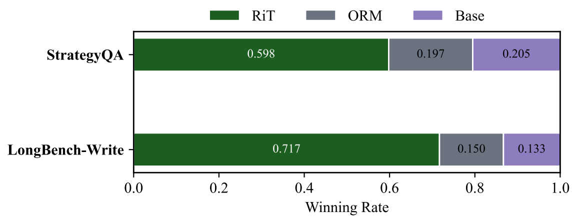

Figure 6: Comparison of the thinking quality of Qwen3-8B under three strategies (Base, ORM, and RiT) on LongBench-Write and StrategyQA. A frontier LLM selects the best reasoning trajectory for each query, and the fraction of selections (i.e., the winning rate) is used as a proxy for the thinking quality.

5.6 Evaluation Robustness

We further evaluate the robustness of our method.

Multiple-judge evaluation. For LongBench-Write and WritingBench, we re-evaluate Qwen3-8B-based models using five judge models: Qwen3-235B-A22B-Instruct, GPT-4o (OpenAI, 2024), Gemini 2.5 Pro (Team, 2025), Claude Sonnet 4, and GPT-5.2 (OpenAI, 2025). As shown in Table 4, RiT consistently achieves the best performance across all judges on both benchmarks, further demonstrating that the observed improvements remain robust across different evaluators.

Multi-sample evaluation. We additionally report results under multiple sampled responses. For reasoning tasks, we report pass@1, pass@8, and avg@8, where pass@k indicates whether at least one of the k sampled responses is correct, and avg@8 denotes the average metric over 8 sampled responses. For open-ended generation tasks, we report pass@1, max@8, and avg@8, where max@8 denotes best-of-8 performance and avg@8 measures the average quality across 8 sampled responses. For avg@8, we additionally report the standard deviation over repeated sampling runs.

As shown in Table 3, RiT consistently outperforms all baselines across all evaluation metrics.

Judge Model	LongBench-Write			WritingBench		
	Base	ORM	RiT	Base	ORM	RiT
Qwen3-235B-A22B-Instruct	84.40	85.70	**89.20**	7.45	8.10	**8.24**
GPT-4o	88.97	85.48	**90.64**	7.23	7.55	**7.64**
Gemini-2.5-Pro	80.57	78.50	**83.94**	6.71	7.62	**7.80**
Claude Sonnet 4	78.66	79.09	**81.89**	6.51	7.13	**7.24**
GPT-5.2	76.30	72.94	**79.32**	6.15	6.47	**6.54**

Table 4: Results across multiple LLM judges on LongBench-Write and WritingBench.

5.7 Results on Mathematical Reasoning Tasks

We further evaluate RiT on **AIME 2024** (Mathematical Association of America, 2024) and **AIME 2025** (Mathematical Association of America, 2025), two challenging competition-level mathematical reasoning benchmarks that require multi-step problem solving and strong symbolic reasoning abilities. All methods are trained on the same dataset curated from AIME questions and answers spanning 1983 to 2023. The results are reported in Table 5. We report both pass@1 and pass@8 to evaluate model performance under single-sample and multi-sample inference settings, respectively. Overall, RiT achieves the best performance on both benchmarks, demonstrating consistent improvements over all baselines. These results are likely attributed to the supervision from the thinking rubric, which encourages more structured and effective reasoning.

Method	AIME 2024		AIME 2025	
	pass@1	pass@8	pass@1	pass@8
Base	0.700	0.758	**0.667**	0.667
ORM	0.733	0.721	0.600	0.667
RiT	**0.767**	**0.762**	**0.667**	**0.675**

Table 5: Comparison of methods based on Qwen3-8B across AIME 2024 and AIME 2025.

5.8 Case Study

We conduct case studies to qualitatively assess the impact of thinking-rubric supervision, with a representative example from LongBench-Write shown in Figure 7 (additional cases are provided in Appendix A.5). In a task requiring a 1,200-word generation,ORM fails to regulate content expansion, resulting in a 2,119-word output, which substantially exceeds the target length. In contrast, RiT produces a more aligned 1,328-word response, closer to the required constraint. This precision stems directly from the length-planning dimension within our thinking rubrics: RiT 's thinking process explicitly pre-allocates word budgets to each subsection. This demonstrates that under thinking rubrics guided supervision of intermediate planning, RiT fosters the disciplined, constraint-aware reasoning necessary for complex, open-ended tasks.

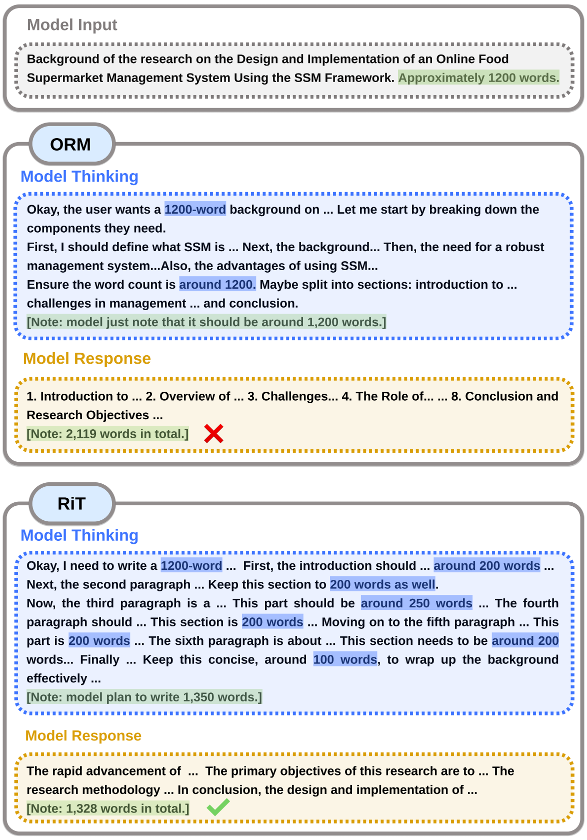

Figure 7: A case study on open-ended generation showing that ORM exceeds the target length due to coarse planning, while RiT meets the length requirement by explicitly planning word budgets during thinking.

6 Conclusion

In this paper, we propose RiT, the first framework to incorporate rubric-based supervision into intermediate reasoning, enabling direct evaluation and optimization of thinking trajectories. RiT automatically generates fine-grained thinking rubrics using and assigns thinking-level rewards based on rubric compliance. To balance reasoning quality with final-task performance, we integrate the thinking reward with an outcome-based reward through a gated fusion mechanism, ensuring consistency between intermediate reasoning and final responses. Experiments demonstrate that RiT improves reasoning quality across diverse tasks, highlighting the value of thinking-rubrics as a scalable and flexible framework for guiding human-aligned reasoning.

Limitations

Despite the effectiveness of RiT, several limitations remain. First, the generation of thinking rubrics relies automated processes with minimal human oversight, which may limit their alignment with nuanced human expectations or domain-specific knowledge. Incorporating human-in-the-loop verification could help mitigate this issue in future work. Second, the integration of thinking-rubric and response rewards currently requires manual hyperparameter tuning. Future work could explore adaptive mechanisms to dynamically balance these reward scales across tasks of varying complexity. Finally, due to computational constraints, experiments were conducted on moderately sized models (e.g., 8B parameters), leaving the scalability and emergent behaviors of RiT on ultra-large models (e.g., >70B parameters) to be investigated.

References

Anthropic. 2025. Introducing claude 4. `https://www.anthropic.com/news/claude-4`.

Ge Bai, Jie Liu, Xingyuan Bu, Yancheng He, Jiaheng Liu, Zhanhui Zhou, Zhuoran Lin, Wenbo Su, Tiezheng Ge, Bo Zheng, and Wanli Ouyang. 2024. Mt-bench-101: A fine-grained benchmark for evaluating large language models in multi-turn dialogues. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 7421–7454. Association for Computational Linguistics.

Yushi Bai, Jiajie Zhang, Xin Lv, Linzhi Zheng, Siqi Zhu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2025. Longwriter: Unleashing 10, 000+ word generation from long context llms. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. Evaluating large language models trained on code. *CoRR*, abs/2107.03374.

Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. 2025. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *CoRR*, abs/2503.09567.

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

Ahmed El-Kishky, Alexander Wei, Andre Saraiva, Borys Minaiev, Daniel Selsam, David Dohan, Francis Song, Hunter Lightman, Ignasi Clavera Gilaberte, Jakub Pachocki, Jerry Tworek, Lorenz Kuhn, Lukasz Kaiser, Mark Chen, Max Schwarzer, Mostafa Rohaninejad, Nat McAleese, o3 contributors, Oleg Mürk, and 5 others. 2025. Competitive programming with large reasoning models. *CoRR*, abs/2502.06807.

Lishui Fan, Yu Zhang, Mouxiang Chen, and Zhongxin Liu. 2025. Posterior-grpo: Rewarding reasoning processes in code generation. *CoRR*, abs/2508.05170.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? A question answering benchmark with implicit reasoning strategies. *Trans. Assoc. Comput. Linguistics*, 9:346–361.

Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Yuanzhuo Wang, and Jian Guo. 2024. A survey on llm-as-a-judge. *CoRR*, abs/2411.15594.

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

Zenan Huang, Yihong Zhuang, Guoshan Lu, Zeyu Qin, Haokai Xu, Tianyu Zhao, Ru Peng, Jiaqi Hu, Zhanming Shen, Xiaomeng Hu, Xijun Gu, Peiyi Tu, Jiaxin Liu, Wenyu Chen, Yuzhuo Fu, Zhiting Fan, Yanmei Gu, Yuanyuan Wang, Zhengkai Yang, and 2 others. 2025. Reinforcement learning with rubric anchors. *CoRR*, abs/2508.12790.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, and 80 others. 2024. Openai o1 system card. *CoRR*, abs/2412.16720.

Ruipeng Jia, Yunyi Yang, Yongbo Gai, Kai Luo, Shihao Huang, Jianhe Lin, Xiaoxi Jiang, and Guanjun Jiang. 2025. Writing-zero: Bridge the gap between non-verifiable tasks and verifiable rewards. *CoRR*, abs/2506.00103.

Muhammad Khalifa, Rishabh Agarwal, Lajanugen Logeswaran, Jaekyeom Kim, Hao Peng, Moontae Lee, Honglak Lee, and Lu Wang. 2025. Process reward models that think. *CoRR*, abs/2504.16828.

Jinu Lee and Julia Hockenmaier. 2025. Evaluating step-by-step reasoning traces: A survey. *CoRR*, abs/2502.12289.

Bill Yuchen Lin, Yuntian Deng, Khyathi Raghavi Chandu, Abhilasha Ravichander, Valentina Pyatkin, Nouha Dziri, Ronan Le Bras, and Yejin Choi. 2025. Wildbench: Benchmarking llms with challenging tasks from real users in the wild. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 3214–3252. Association for Computational Linguistics.

Hanmeng Liu, Jian Liu, Leyang Cui, Zhiyang Teng, Nan Duan, Ming Zhou, and Yue Zhang. 2023a. Logiqa 2.0 - an improved dataset for logical reasoning in natural language understanding. *IEEE ACM Trans. Audio Speech Lang. Process.*, 31:2947–2962.

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

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023b. G-eval: NLG evaluation using gpt-4 with better human alignment. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing, EMNLP 2023, Singapore, December 6-10, 2023*, pages 2511–2522. Association for Computational Linguistics.

Yifei Lu, Fanghua Ye, Jian Li, Qiang Gao, Cheng Liu, Haibo Luo, Nan Du, Xiaolong Li, and Feiliang Ren. 2025. Codetool: Enhancing programmatic tool invocation of llms via process supervision. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 18287–18304. Association for Computational Linguistics.

Xinji Mai, Haotian Xu, Xing W, Weinong Wang, Yingying Zhang, and Wenqiang Zhang. 2025. Agent RL scaling law: Agent RL with spontaneous code execution for mathematical problem solving. *CoRR*, abs/2505.07773.

Sara Vera Marjanovic, Arkil Patel, Vaibhav Adlakha, Milad Aghajohari, Parishad BehnamGhader, Mehar Bhatia, Aditi Khandelwal, Austin Kraft, Benno Krojer, Xing Han Lù, Nicholas Meade, Dongchan Shin, Amirhossein Kazemnejad, Gaurav Kamath, Marius Mosbach, Karolina Stanczak, and Siva Reddy. 2025. Deepseek-r1 thoughtology: Let's about LLM reasoning. *CoRR*, abs/2504.07128.

Mathematical Association of America. 2024. American invitational mathematics examination (aime), 2024 edition. `https://maa.org/maa-invitational-competitions/`. Accessed: 2026-04-14.

Mathematical Association of America. 2025. American invitational mathematics examination (aime), 2025 edition. `https://maa.org/maa-invitational-competitions/`. Accessed: 2026-04-14.

OpenAI. 2024. Gpt-4o system card. *CoRR*, abs/2410.21276.

OpenAI. 2025. Introducing gpt-5.2. `https://openai.com/index/introducing-gpt-5-2/`.

Hao Peng, Yunjia Qi, Xiaozhi Wang, Zijun Yao, Bin Xu, Lei Hou, and Juanzi Li. 2025. Agentic reward modeling: Integrating human preferences with verifiable correctness signals for reliable reward systems. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 15934–15949. Association for Computational Linguistics.

Chau Pham, Simeng Sun, and Mohit Iyyer. 2024. Suri: Multi-constraint instruction following in long-form text generation. In *Findings of the Association for Computational Linguistics: EMNLP 2024, Miami, Florida, USA, November 12-16, 2024*, pages 1722–1753. Association for Computational Linguistics.

Haoran Que, Feiyu Duan, Liqun He, Yutao Mou, Wangchunshu Zhou, Jiaheng Liu, Wenge Rong, Zekun Moore Wang, Jian Yang, Ge Zhang, Junran Peng, Zhaoxiang Zhang, Songyang Zhang, and Kai Chen. 2024. Hellobench: Evaluating long text generation capabilities of large language models. *CoRR*, abs/2409.16191.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347.

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

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. 2025. Hybridflow: A flexible and efficient RLHF framework. In *Proceedings of the Twentieth European Conference on Computer Systems, EuroSys 2025, Rotterdam, The Netherlands, 30 March 2025 - 3 April 2025*, pages 1279–1297. ACM.

Gemini Team. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *CoRR*, abs/2507.06261.

Haozhe Wang, Haoran Que, Qixin Xu, Minghao Liu, Wangchunshu Zhou, Jiazhan Feng, Wanjun Zhong, Wei Ye, Tong Yang, Wenhao Huang, Ge Zhang, and Fangzhen Lin. 2025a. Reverse-engineered reasoning for open-ended generation. *CoRR*, abs/2509.06160.

Pengkai Wang, Qi Zuo, Pengwei Liu, Zhijie Sang, Congkai Xie, and Hongxia Yang. 2025b. Infimed-orbit: Aligning llms on open-ended complex tasks via rubric-based incremental training. *CoRR*, abs/2510.15859.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Yuning Wu, Jiahao Mei, Ming Yan, Chenliang Li, Shaopeng Lai, Yuran Ren, Zijia Wang, Ji Zhang, Mengyue Wu, Qin Jin, and Fei Huang. 2025. Writingbench: A comprehensive benchmark for generative writing. *CoRR*, abs/2503.05244.

Wei Xiong, Wenting Zhao, Weizhe Yuan, Olga Golovneva, Tong Zhang, Jason Weston, and Sainbayar Sukhbaatar. 2025. Stepwiser: Stepwise generative judges for wiser reasoning. *CoRR*, abs/2508.19229.

Fengli Xu, Qianyue Hao, Zefang Zong, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, Jiahui Gong, Tianjian Ouyang, Fanjin Meng, Chenyang Shao, Yuwei Yan, Qinglong Yang, Yiwen Song, Sijian Ren, Xinyuan Hu, Yu Li, Jie Feng, Chen Gao, and Yong Li. 2025. Towards large reasoning models: A survey of reinforced reasoning with large language models. *CoRR*, abs/2501.09686.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 40 others. 2025. Qwen3 technical report. *CoRR*, abs/2505.09388.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, and 16 others. 2025a. DAPO: an open-source LLM reinforcement learning system at scale. *CoRR*, abs/2503.14476.

Zhuohao Yu, Weizheng Gu, Yidong Wang, Xingru Jiang, Zhengran Zeng, Jindong Wang, Wei Ye, and Shikun Zhang. 2025b. Reasoning through execution: Unifying process and outcome rewards for code generation. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*. OpenReview.net.

Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, Jingren Zhou, and Junyang Lin. 2025. Group sequence policy optimization. *CoRR*, abs/2507.18071.

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

A Details of Experiments

A.1 Implementation Details

We train our models using the VeRL framework[2] (Sheng et al., 2025) and follow standard GRPO hyperparameter settings. Specifically, the learning rate is set to 2×10^{-6}, and each rollout consists of 8 sampled responses. For open-ended generative tasks, we use a batch size of 32, while for reasoning or inference tasks, the batch size is set to 128. The thinking rubrics are generated by Claude Sonnet 4. During training, model outputs are scored by Qwen3-235B-A22B-Instruct. For open-ended generation, the fusion weight α is generally set to 0.5, while for DeepSeek-R1-Distilled-Llama-8B on LongBench-Write, α is set to 1. For reasoning tasks, α is consistently set to 1. Training proceeds until the reward converges. The models are run for around 500 steps on reasoning tasks, and around 200 steps for open-ended generation tasks.

A.2 Datasets

Here we detail the training and test splits for all datasets used in our experiments. Different split ratios (50:50 vs. 80:20) are chosen based on dataset size and evaluation needs, while ensuring that training and test data are drawn from the same underlying distribution. An overview of the splits is provided in Table 6.

- **LongBench-Write** is split 50:50 by type and text length, preserving the original distribution across these attributes to ensure balanced evaluation of long-form generation quality.

- **WritingBench** is split 50:50 via stratified sampling over major/minor domains and language, ensuring representative coverage of all six professional writing domains in both splits.

- **StrategyQA** uses the official benchmark split, with the training set and development set as the test set for comparability with prior work.

- **LogiQA** adopts an NLI task formulation as suggested by (Liu et al., 2023a), focusing on logical inference between premise and hypothesis. We use the benchmark-provided training and test sets to ensure consistency with the standard evaluation protocol.

- **TruthfulQA** uses the multiple-choice 1 (MC1) task format and is split 80:20 into training and test sets. This ratio provides sufficient training data given the limited dataset size while preserving a reliable test set.

Dataset	Training Set	Test Set
LongBench-Write	60	60
WritingBench	500	500
StrategyQA	2,061	229
LogiQA	30,908	3,942
TruthfulQA	653	164

Table 6: Overview of the training and test splits for all datasets used in our experiments.

A.3 Evaluation Protocol

Here we describe the evaluation protocol and scoring procedures used in our experiments.

For LongBench-Write and WritingBench, we use the official evaluation protocols and directly adopt the evaluation scripts released in their public GitHub repositories[3], without any modification to the evaluation prompts or scoring logic.

For LongBench-Write, the final score ranges from 0 to 100 and is computed as the average of a length score (S_l) and a quality score (S_q). For WritingBench, the evaluator independently assigns a 10-point score to the generated response for each quality criterion $c_i \in C_q$, and the final score is computed by averaging across all criteria.

A.4 Impact of Fusion Weight α Choice.

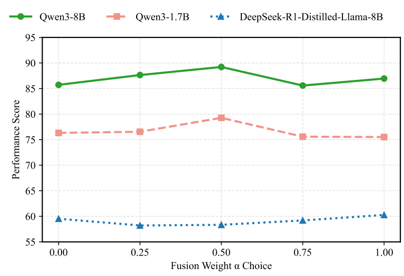

Figure 8: Performance scores of different models at varying fusion weights α in LongBench-Write.

Figure 8 illustrates the performance scores of different models under varying fusion weights α in LongBench-Write. Models with relatively higher

[2] https://github.com/volcengine/verl

[3] https://github.com/THUDM/LongWriter and https://github.com/X-PLUG/WritingBench

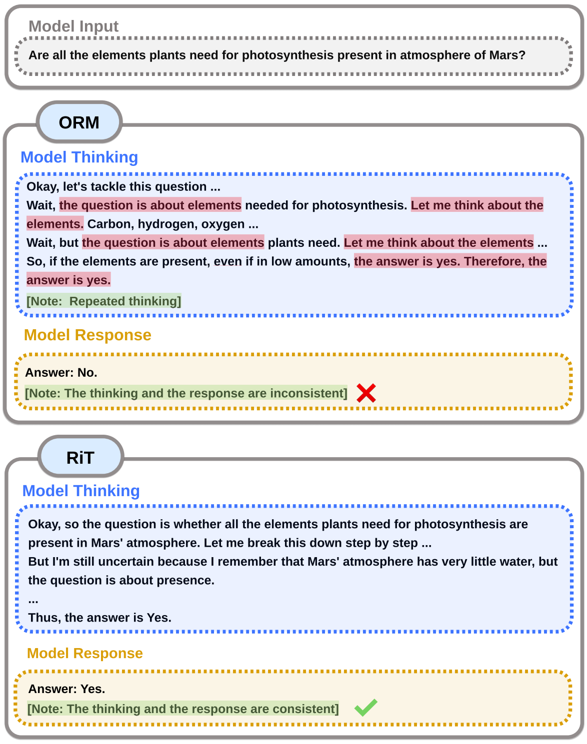

Figure 9: A case study on a reasoning task. ORM exhibits repetitive and unfocused reasoning, producing an incorrect answer, whereas RiT follows a disciplined reasoning process guided by thinking rubrics and arrives at the correct answer.

baseline performance generally tend to achieve their best results at moderate α values, as excessive weighting toward the thinking score can override the model's strong predictions and slightly degrade performance. In contrast, the model with lower baseline performance, such as DeepSeek-R1-Distill-Llama-8B, benefits from higher α, suggesting that the thinking score can compensate for weaker predictions and consequently enhance overall results.

A.5 Case Study

We present additional case studies to illustrate the effectiveness of thinking-rubric supervision in guiding model reasoning and content generation.

WritingBench (Open-Ended Generation). As shown in Figure 10, the task is to write a detailed guidance document for parents, incorporating interview summaries and industry reports. ORM lacks explicit planning for integrating external inputs, resulting in omissions and disorganized content. RiT leverages content-mapping from the thinking rubrics, specifying which information belongs in which section, and produces a coherent, information-complete document.

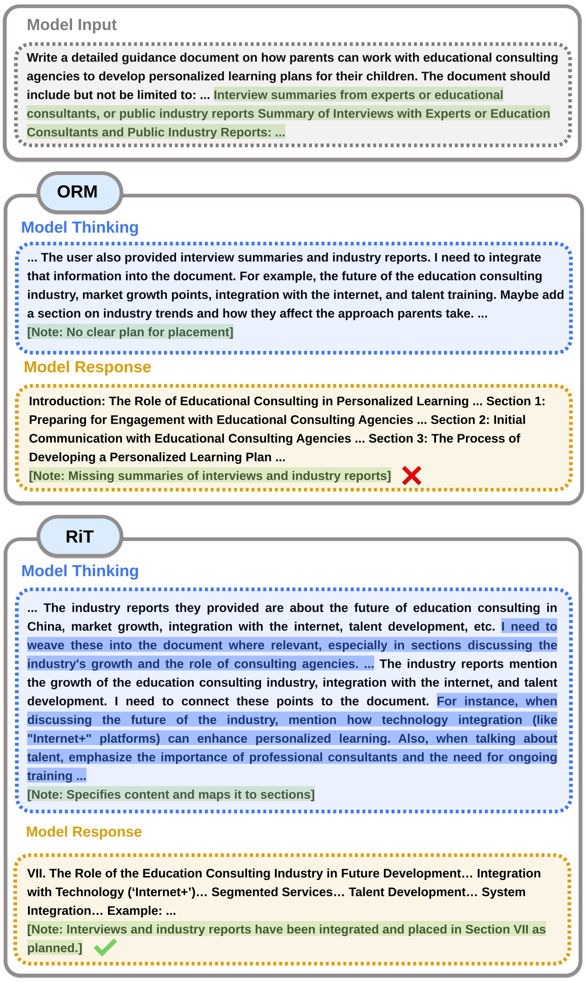

Figure 10: A case study on open-ended generation. ORM lacks structured planning and omits key details, while RiT uses content-mapping from thinking rubrics to map content to sections and produce a coherent document.

StrategyQA (Reasoning Task). As shown in Figure 9, the task asks whether Mars' atmosphere contains all elements needed for photosynthesis. ORM exhibits looping reasoning, repeatedly revisiting similar content and ultimately producing an incorrect answer despite internally concluding correctly. RiT, guided by thinking-rubric supervision, maintains a disciplined reasoning trajectory, prioritizing critical elements and avoiding redundant loops, leading to the correct answer.

Overall, these case studies show thinking-rubric supervision enables RiT to produce more precise, structured, and constraint-aware outputs across reasoning and open-ended generation tasks.