Beyond Length Scaling: Synergizing Breadth and Depth for Generative Reward Models

Qiyuan Zhang[*♠◇], **Yufei Wang**[◇†],
Tianhe Wu[♠], **Can Xu**[◇†], **Qingfeng Sun**[◇], **Kai Zheng**[◇] **Xue Liu**[♡], **Chen Ma**[♠†]
[♠]City University of Hong Kong [◇]Tencent Hunyuan [♡]MBZUAI
qzhang732-c@my.cityu.edu.hk

Abstract

Recent advancements in Generative Reward Models (GRMs) have demonstrated that scaling the length of Chain-of-Thought (CoT) reasoning considerably enhances the reliability of evaluation. However, current works predominantly rely on unstructured length scaling, ignoring the divergent efficacy of different reasoning mechanisms: Breadth-CoT (multi-dimensional principle coverage) and Depth-CoT (substantive judgment soundness). To address this, we introduce **Mix-GRM**, a framework that reconfigures raw rationales into structured Breadth-CoT and Depth-CoT through a modular synthesis pipeline, subsequently employing Supervised Fine-Tuning (SFT) and Reinforcement Learning with Verifiable Rewards (RLVR) to internalize and optimize these mechanisms. Comprehensive experiments demonstrate that Mix-GRM establishes a new state-of-the-art across five benchmarks, surpassing the best open-source baseline by 3.5%. Our results reveal a clear divergence in reasoning: Breadth-CoT benefits subjective preference tasks, whereas Depth-CoT excels in objective correctness tasks. Consequently, misaligning the reasoning mechanism with the task directly degrades performance. Furthermore, we demonstrate that RLVR acts as a switching amplifier, inducing an emergent polarization where the model spontaneously allocates its reasoning style to match task demands. The synthesized data and models are released at Hugging Face, and the code is released at Github

1 Introduction

Reinforcement learning (RL) has proven to be the critical post-training mechanism for eliciting capabilities in Large Language Models (LLMs) (Ouyang et al., 2022; Team, 2025a,b). However, as the ambition of RL expands from single-domain optimization (*e.g.*, math) (Le et al., 2022; Shao et al., 2024; Wang et al., 2025a) to general-purpose alignment (Lee et al., 2024; Shen et al., 2025), the Reward Model (RM) faces the challenge of providing reliable feedback for increasingly complex queries from diverse, real-world scenarios (Liu et al., 2025d; Li et al., 2025a). Addressing this challenge requires a shift in RM design. Inspired by how CoT (Wei et al., 2023; Yeo et al., 2025) trades inference-time compute for enhanced generalization performance, the community has increasingly adopted Generative Reward Models (GRMs) (Zheng et al., 2023; Yuan et al., 2024; Zhang et al., 2025a). By prompting an explicit evaluation rationale prior to conclusion, GRMs aim to transfer the robust generalization observed in CoT generation to the task of reward modeling.

Building on these successes, existing GRM methods predominantly leverage CoT by simply scaling its length (Chen et al., 2025b,a; Zhang et al., 2025c), feeding it with massive evaluation signals, such as fine-grained features (Kim et al., 2024) or multi-perspective critiques (Ankner et al., 2024). However, prior CoT studies (Sprague et al., 2025; Besta et al., 2025; Wang et al., 2024b; Kambhampati et al., 2024) have established that longer CoTs do not universally guarantee performance gains; rather, the optimal structural bias diverges significantly across domains. Crucially, recent insights from test-time scaling (Li et al., 2025b; Zhang et al., 2025b) provide a theory for this divergence, identifying *parallel thinking* and *sequential thinking* as two fundamental, orthogonal mechanisms for amplifying intelligence. Conceptually, reasoning-heavy tasks (*e.g.*, math, code) necessitate sequential verification to ensure deductive rigor (Wang et al., 2024a; Liu et al., 2025a; Lightman et al., 2023), whereas semantic-heavy tasks (*e.g.*, open-ended generation) benefit from parallel exploration to ensure comprehensive coverage of diverse possibilities (Zheng et al., 2025; Pan et al., 2025).

[*]Work done during his internship at Tencent Hunyuan.

Findings of the Association for Computational Linguistics: ACL 2026, pages 14449–14469

Drawing on this distinction, we argue that advancing RM requires shifting focus from merely scaling CoT length to aligning its reasoning mechanisms with task demands. Specifically, this necessitates a transition from static, one-size-fits-all CoT templates toward a *mix reasoning mechanism*. Thus, we propose **Mix-GRM**, which implements a dynamic mix reasoning mechanism within a unified reward modeling framework. Specifically, we introduce a synthesis framework that reconfigures raw, unstructured rationales into two distinct long CoTs: Breadth-CoT (B-CoT) and Depth-CoT (D-CoT). To achieve this, we first decouple unstructured rationales into atomic "Principle–Judgment–Verdict" units. This modularity allows us to reassemble the units into syntactically unified but structurally diverse paths. To illustrate, a B-CoT is synthesized by the parallel aggregation of units across diverse principles (*e.g.*, combining an 'Accuracy' unit with a 'Clarity' unit) to ensure coverage. Conversely, D-CoT extends the CoT by first performing a direct reasoning pass to solve the instruction, thereby enabling a re-evaluated judgment grounded in the generated reasoning pass to ensure soundness. To cultivate mechanism-adaptive alignment, we construct a synergistic mixture dataset by pairing B-CoT with subjective preference tasks and D-CoT with objective correctness tasks. We first initialize the model via SFT on this mixture and subsequently optimize it through RLVR using normal RM datasets, where only final labels are available.

Comprehensive experiments across five standard benchmarks yield three critical conclusions: (1) **Universal SOTA Performance and Downstream Utility**: *Mix-GRM* establishes a new state-of-the-art, consistently surpassing strong baselines like *Skywork-Reward* and *FARE-8B* on general reward benchmarks. Crucially, this superiority extends to practical downstream tasks: *Mix-GRM* demonstrates best-in-class utility in both Offline RL (DPO) and Test-time Scaling (Best-of-N). (2) **Divergent Roles of Reasoning Mechanisms**: Our analysis reveals that B-CoT predominantly benefits subjective preference but degrades objective correctness, while D-CoT excels in correctness at the cost of preference. This confirms that the efficacy of a reasoning mechanism is task-dependent. (3) **RLVR as a Switching Amplifier**: Mixed mechanisms provide a superior base for RL. RLVR boosts *Mix-GRM* by a larger margin than the *Base-GRM*. Our analysis demonstrates that RLVR automatically sharpens the mechanism allocation—spontaneously converging on B-CoT for preference and D-CoT for correctness. This confirms that optimizing how a model thinks is more critical for post-training efficacy than simply scaling how long it writes.

2 Related Work

2.1 Generative Reward Model

Generative Reward Models represent a paradigm shift from scalar regression to explicit reasoning. Developing alongside the prompting-based "LLM-as-a-Judge" paradigm (Zheng et al., 2023), GRMs are explicitly trained to generate natural language rationales alongside preference decisions (Yuan et al., 2024). Driven by the transformative success of long CoT, the research trajectory in this field has pivoted toward continuously extending the length of these rationales. To achieve this, many work leverages RL to explicitly elicit and stabilize longer CoT traces (Chen et al., 2025b,a; Whitehouse et al., 2025), while complementary efforts utilize detailed rubrics/checklists to synthetically expand evaluation coverage (Kim et al., 2024; Liu et al., 2025b; Gunjal et al., 2025; Viswanathan et al., 2025). However, while these strategies successfully scale the quantity of reasoning, they typically rely on static, task-agnostic structures, overlooking the critical nuance that the optimal reasoning mechanism is intrinsically task-dependent.

2.2 Breadth and Depth in Chain-of-Thought

The evolution of CoT is fundamentally characterized by the continuous exploration of diverse structures (Shinn et al., 2023; Team, 2025a). Beyond simple linear chains, frameworks such as Tree of Thoughts (Yao et al., 2023) and Graph of Thoughts (Besta et al., 2025) introduce branching and recurrent topologies, framing reasoning as a structured search over partial thoughts. Complementing these complex structures, approaches like Skeleton-of-Thought (Ning et al., 2024) and Self-Consistency (Wang et al., 2023) demonstrate the efficacy of parallel exploration, leveraging lateral breadth to enhance robustness and coverage. Collectively, these studies establish that reasoning is not structure-agnostic; rather, specific topological priors—ranging from deep sequential trees to broad parallel ensembles—are required to unlock optimal performance across distinct domains (Sprague et al., 2025), a distinction that our work formally adapts to reward modeling.

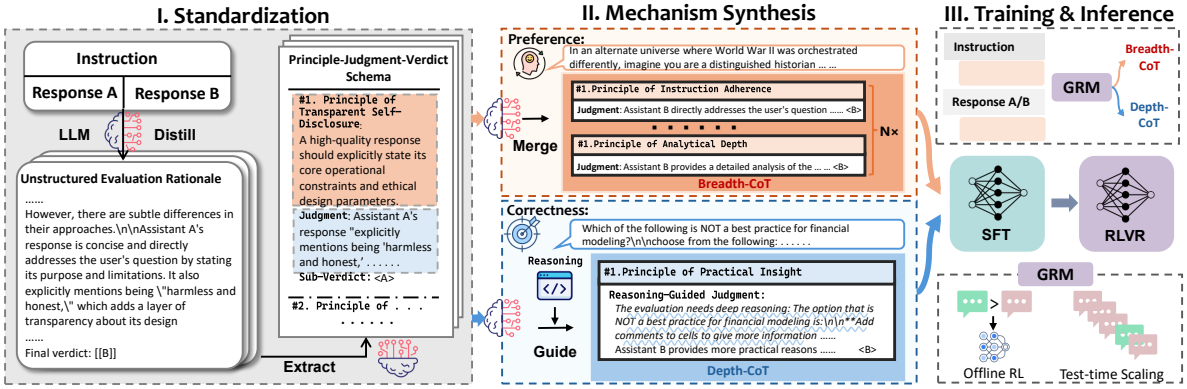

Figure 1: The pipeline of the Mix-GRM. (i) **Standardization**: We extract raw rationales into modular *Principle–Judgment–Verdict* units. (II) **Mechanism Synthesis**: We reconstruct modules into *Breadth-CoT* for preference or *Depth-CoT* for correctness. (III) **Training & Inference**: Following SFT and RLVR training, the model achieves mechanism-adaptive alignment, automatically deploying the optimal mechanism for inference and providing reliable signals for downstream tasks like Offline RL and test-time scaling.

3 Methodology

We propose the Mix-GRM, a framework designed to dynamically align the reasoning mechanism with intrinsic task demands. Moving beyond static, unstructured rationale sequences, our approach formalizes evaluation into two orthogonal CoTs: **B-CoT**, which enforces the lateral aggregation of diverse principles, and **D-CoT**, which necessitates the expansion of judgment. As illustrated in Figure 1, our methodology comprises three key phases: modular schema standardization (§3.2), mechanism synthesis (§3.3), and mechanism-adaptive alignment (§3.4).

3.1 Problem Formulation

Supposing $\{y_A, y_B\}$ denote two candidate responses generated by two assistants A and B for a given task instruction x, a normal GRM $\mathcal{M}$ produces an output sequence consisting of an explicit evaluation rationale c followed by a preference verdict v, comparing the quality of y_A and y_B.

$$(c, v) = \mathcal{M}(y_A, y_B \mid x).$$

The objective is to ensure that the v aligns with human preference. In our framework, we denote the full input triplet as $I = (x, y_A, y_B)$.

3.2 Modular Schema Standardization

Conventional GRMs typically produce the rationale c as an unstructured, free-form sequence. Inspired by recent checklist-based evaluation (Viswanathan et al., 2025), which advocates for the atomization of the complex evaluation process into checklist-driven points, we propose to reconfigure these raw rationales into a structured "Principle-Judgment-Verdict" Schema (Figure 1, Stage I). By transforming tangled rationales into atomic units, we ensure that the RM's reasoning process is both interpretable and granularly verifiable. Formally, we utilize a LLM to parse the raw c into structured atomic units $\mathcal{S}$:

$$\mathcal{S} = \{(p_k, j_k, v_k)\}_{k=1}^{K},$$

where p_k denotes a discrete evaluation **Principle** (*e.g.*, "Instruction Adherence"), j_k represents the specific **Judgment** (*e.g.*, "Response B directly addresses...") analyzing that principle, and v_k is the following **Sub-Verdict** (*e.g.*, "<B> is Better"). Here, K typically ranges from 3 to 5.

This atomic decomposition yields cleaner learning signals and ensures syntactic uniformity (Li et al., 2025c), ensuring that performance gains are driven by thinking mechanisms (*i.e.*, Breadth vs. Depth) rather than superficial stylistic patterns.

3.3 Mechanism Synthesis

Building on the $\mathcal{S}$, we introduce a dual-track synthesis pipeline (Figure 1, Stage II) to synthesize **B-** and **D-CoT** as follows:

B-CoT Synthesis. We define B-CoT as the parallel aggregation of distinct principles, designed to overcome the narrow focus of single-pass rationale. In subjective preference tasks, where a "good" response is defined by the simultaneous satisfaction of multi-dimensional factors (*e.g.*, tone, helpfulness, and creativity), single-track reasoning often fixates on dominant traits while overlooking subtle, fine-grained details. By exploring diverse

reasoning paths concurrently, parallel thinking provides a deliberative breadth that aligns with the multifaceted nature of human preference. To simulate parallel thinking, we treat independent sampling as a stochastic exploration of the instruction's evaluative manifold. By sampling N independent rationales $\{c_n\}_{n=1}^{N}$ from multiple cognitive trajectories, we elicit a diverse set of hidden principles that might otherwise remain dormant. These rationales are parsed into structured schemas $\{\mathcal{S}_n\}$ and subsequently unified via an LLM-based **Merge & Deduplicate** transformation $\mathcal{T}_{\text{merge}}$:

$$C_{\text{breadth}} = \mathcal{T}\text{merge}\left(\bigcup n = 1^{N}(p, j, v) \in \mathcal{S}_n\right).$$

Here, we filter out lowest-frequency principles. This synthesis yields a comprehensive, non-redundant spectrum of principles, effectively expanding the model's horizontal evaluative scope.

D-CoT Synthesis. We define D-CoT as the expansion of judgment to ensure substantive reasoning soundness by mitigating superficial shortcuts. In contrast to subjective preferences, a "good" response in objective correctness tasks depends on rigorous logical constraints (*e.g.*, mathematical proofs or functional code). Normal rationales often fixate on surface-level fluency (*e.g.*, professional tone or formatting) while failing to verify the underlying logical validity. By enforcing the sequential verification of logical dependencies, sequential thinking provides a deductive rigor that naturally aligns with the strict requirements of objective correctness. To simulate sequential thinking, we first elicit a Reasoning Trace z—a self-solving pass derived from x that explicitly outlines the optimal solution paths required for a correct response. Recognizing that depth-oriented reasoning demands higher cognitive load per unit, we intentionally trade off horizontal coverage for deductive rigor by sampling a focused subset $\mathcal{S}_{\text{sub}} \subset \mathcal{S}$ (typically $|K| \leq 3$). In this stage of **Reasoning-Guided Judgment**, each unit's judgment is re-generated as a derivative of the trace z:

$$\tilde{j}_k = \mathcal{T}\text{refine}(p_k \mid z)$$

To ensure the evaluative process is transparent and explicitly grounded in the model's own logic, we inject z directly into the lead unit $\tilde{j}_1$. The final C_{depth} is constructed by serializing these refined units, transforming the verdict into a substantive analytical process anchored by the trace z.

3.4 Mechanism-Adaptive Alignment

Training proceeds in two stages (Figure 1, Panel III): SFT on mixture CoT datasets, followed by GRPO (Shao et al., 2024) to align verdicts with human labels.

SFT. Following Frick et al. (2025), we categorize general RM training data into two domains: **Preference** (subjective) and **Correctness** (objective). We construct the mixture dataset $\mathcal{D}_{\text{mix}}$ by assigning C_{breadth} to preference instances and C_{depth} to correctness instances. We first initialize the policy π_θ via SFT on $\mathcal{D}_{mix}$. Given the I, the model is trained to generate the corresponding CoT $c \in \{c_{\text{breadth}}, c_{\text{depth}}\}$ alongside the verdict v.

RLVR via GRPO. To optimize verdict accuracy, we employ RLVR via GRPO (Shao et al., 2024), rewarding the model solely for consistency with ground-truth labels:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{\substack{I\sim\mathcal{D} \\ \{o_i\}\sim\pi_{\theta_{\text{old}}}}}\left[\frac{1}{G}\sum_{i=1}^{G}\left(\frac{\pi_\theta(o_i|I)}{\pi_{\theta_{\text{old}}}(o_i|I)}\hat{A}_i - \beta\mathbb{D}_{\text{KL}}(\pi_\theta||\pi_{\text{ref}})\right)\right]$$

The reward is defined by verdict consistency: a positive reward is assigned if the generated verdict v_i matches the ground-truth human label, and -1 otherwise. This process acts as a **switching amplifier**, inducing an emergent polarization: the model spontaneously learns to couple B-CoT with preference tasks and D-CoT with correctness tasks to maximize rewards, as empirically verified in §5. This confirms that the model autonomously converges on the optimal thinking style for each domain.

4 Experiment

We evaluate Mix-GRM across three objectives: (1) **Overall Performance** against SoTA baselines; (2) **Mechanism Efficiency** to quantify the domain-specific benefits of B- and D-CoT; and (3) **Downstream Utility** in Offline RL and Test-time Scaling.

4.1 Experimental Setup

General Reward Benchmarks. We employ five widely recognized benchmarks tailored for general-purpose reward modeling: RewardBench (Lambert et al., 2024), RewardBench-v2 (Malik et al., 2025), RMB (Zhou et al., 2025a), RM-Bench (Liu et al., 2025c), and PPE (Frick et al., 2025). These

Models	Stage	Data	Benchmarks					
			RB-v1†	RB-v2†	RM-BENCH	RMB	PPE	**Avg.**
Reference: Proprietary Models								
DeepSeek-V3.2	–	–	95.5	92.1	91.4	83.9	69.0	86.4
Gemini-3-Flash	–	–	95.3	91.1	93.8	79.2	76.4	87.2
Open-Source Baselines								
Skywork-Reward-8B	BT	44K	**93.9**	**79.7**	72.4	74.4	61.7	76.5
JudgeLRM-7B	RL	100K	79.0	55.6	78.5	73.1	57.9	68.8
RM-R1-7B (Distill)	SFT , RL	9K , 64K	83.5	48.7	76.6	65.1	62.0	67.2
RM-R1-7B (Instruct)	SFT , RL	9K , 64K	82.3	61.4	75.1	69.9	62.0	70.1
FARE-8B	SFT	2.5M	86.3	73.4	74.1	**83.2**	62.5	75.9
RubricRM-8B	SFT	36K	86.7	71.9	74.0	78.5	62.5	74.7
DeepSeek-GRM-16B	SFT , RL	1.2M , 237K	76.8	56.0	63.5	70.8	59.1	65.2
Ours: Mix-GRMs								
Stage I: SFT-trained								
Base-GRM	SFT	9K	84.5	64.7	77.0	79.2	61.1	73.3
Mix-GRM (Ours)	SFT	9K	87.2	67.8	<u>79.2</u>	78.9	62.1	75.1
Stage II: RLVR-trained								
Base-GRM	SFT , RL	9K , 21K	89.0	74.0	78.8	78.5	<u>64.0</u>	<u>76.9</u>
Mix-GRM (Ours)	SFT , RL	9K , 21K	<u>91.8</u>	<u>77.5</u>	**82.7**	<u>80.1</u>	**64.8**	**79.4**

Table 1: Performance of RMs on reward benchmarks. Among open-source models, the highest score per column is **bolded**, and the second-highest is <u>underlined</u>. "Overall" denotes the average score within each benchmark. Proprietary LLMs (gray rows) are included for reference. †RB-v1/v2 refers to RewardBench v1 and v2.

benchmarks encompass a broad spectrum of tasks, ranging from common tasks like math, coding, and open-ended chat, to specialized capabilities including factuality and instruction-following. For Overall Performance, we report standard benchmark-level pairwise comparison accuracy to assess the rewarding capability. For granular Mechanism Efficiency analysis, we aggregate instances from these benchmarks and re-categorize them into two fundamental domains, Correctness and Preference, based on their original task metadata. Detailed statistics and specific domain mappings for these benchmarks are provided in Appendix B.3.

Base Model and Training Data Source. We employ Qwen3-8B-Base (Team, 2025c) trained on a composite corpus 30,000 samples (9K SFT, 21K RLVR) spanning five datasets: `HelpSteer3` (Wang et al., 2025b) (chat, stem & multilingual), `Code-Preference` (coding), `Math-DPO` (math), `WildGuard` (Han et al., 2024) (safety), and `OffsetBias` (Park et al., 2024) (instruction following). Detailed sampling protocols and statistical distributions are provided in Appendix B.2. Other Training Implementation Setting is in Sec. B.

Baselines. We compare our proposed RM with 7 top-tier RMs across two paradigms: (1) *Discriminative*: represented by **Skywork-Reward-v0.2-Llama-3.1-8B** (Liu et al., 2024), a leading scalar model trained via Bradley-Terry modeling; and (2) *Generative*: encompassing RL-driven reasoning models (**JudgeLRM-7B** (Chen et al., 2025a), **RM-R1-Instruct** (Chen et al., 2025b), **RM-R1-Distill**, **DeepSeek-GRM-16B**) (Liu et al., 2025d), synthetic scaling methods (**FARE-8B** (Xu et al., 2025)), and rubric-based approaches **RubricRM-8B** (Liu et al., 2025b). Notably, RubricRM-8B incorporates two-stage LLMs consisting of a rubric generator and a rubric-based judge.

4.2 Overall Performance in Benchmarks

Table 1 validates the effectiveness of our Mix-GRM through three dimensions.

Effectiveness of Mixture SFT : Via mixture SFT alone, *Mix-GRM* achieves a remarkable average score of 75.1. This performance surpasses GRMs requiring computationally expensive RL to elicit long-CoT capabilities—outperforming *RM-R1-Instruct* by 5.0 and *DeepSeek-GRM-16B* by 9.9. Furthermore, it beats *RubricRM-8B* (+0.4), which relies on a complex but static rubric-template CoT. This confirms that aligning reasoning mechanisms serves as a potent alternative strategy, alongside approaches focused on RL exploration or static template engineering.

Models	Preference Domain						Correctness Domain						Overall
	RB-v1	RB-v2	RM-B†	RMB	PPE	**Avg.**	RB-v1	RB-v2	RM-B†	RMB	PPE	**Avg.**	
Baselines													
FARE-8B	85.0	57.3	66.9	**82.9**	59.6	70.4	85.2	67.3	63.0	88.1	63.3	73.3	71.9
RubricRM-8B	82.4	56.0	62.2	77.5	**64.9**	68.6	87.6	64.2	57.6	86.5	60.4	71.3	70.0
DeepSeek-GRM	80.6	59.6	64.0	76.8	59.8	68.2	76.6	55.8	56.6	86.8	56.8	66.5	67.4
Ours: Mix-GRMs													
Stage I: SFT-trained													
Base-GRM	81.6	55.5	63.3	80.5	60.1	68.2	84.1	63.7	67.7	86.4	59.1	72.2	70.2
Mix-GRM (Breadth)	83.7	59.1	65.9	77.9	59.5	69.3$_{\uparrow 1.1}$	81.1	60.2	64.1	86.8	58.7	70.2$_{\downarrow 2.0}$	69.8
Mix-GRM (Depth)	80.3	50.2	70.6	70.1	58.6	65.9$_{\downarrow 2.3}$	88.0	63.7	66.7	81.1	64.7	72.8$_{\uparrow 0.6}$	69.4
Mix-GRM	84.9	55.7	71.2	78.7	59.2	70.0$_{\uparrow 1.8}$	88.4	65.8	67.7	81.9	63.7	73.5$_{\uparrow 1.3}$	71.8
Stage II: RLVR-trained													
Base-GRM	83.0	58.0	68.5	73.8	61.4	68.9$_{\uparrow 0.7}$	89.8	69.5	69.9	**89.5**	63.4	76.4$_{\uparrow 4.2}$	72.7
Mix-GRM (Breadth)	**86.2**	58.8	70.1	79.2	60.7	71.0$_{\uparrow 2.8}$	82.8	63.4	64.3	86.5	60.7	71.5$_{\downarrow 0.7}$	71.3
Mix-GRM (Depth)	85.2	57.8	**75.6**	75.4	61.2	**71.0**$_{\uparrow 2.8}$	91.8	70.3	72.9	87.4	**66.2**	77.7$_{\uparrow 5.5}$	74.4
Mix-GRM	**86.2**	**64.4**	72.7	78.1	61.7	**72.6**$_{\uparrow 3.7}$	**92.2**	**72.5**	**74.5**	88.9	65.4	**78.7**$_{\uparrow 6.5}$	**75.7**

Table 2: Performance of RMs grouped by domain. "Avg." denotes the domain average. We annotate the performance gap relative to the *Base-GRM in SFT* baseline within the same stage using colored subscripts (↑ for gain, ↓ for drop). Highest score per column is **bolded**, second-highest is underlined. †RM-B refers to RM-Bench.

Models	Instruction-Following			Mathematical Reasoning				
	AlpacaEval-v2	Arena-Hard	**Avg.**	GSM8K	MATH	STEM	TabMWP	**Avg.**
SFT	6.4	4.2	5.3	75.1	25.2	38.6	40.9	45.0
DPO Training (Different RMs)								
↪ RubricRM-8B	8.5	12.5	10.5	76.0	26.9	**41.4**	38.8	45.9
↪ FARE-8B	8.9	**15.1**	12.0	75.7	26.9	39.0	41.4	45.8
↪ RM-R1-Instruct	7.9	14.3	11.1	76.3	26.5	38.5	41.7	45.8
↪ DeepSeek-GRM-16B	8.0	14.1	11.1	75.6	26.6	38.7	41.6	45.6
↪ **Ours (Mix-GRM)**	**9.2**	15.0	**12.1**	**77.6**	**27.1**	39.0	**41.9**	**46.4**

Table 3: Performance of DPO-trained policy models using different reward models on instruction-following and math-reasoning benchmarks. "Avg." is the average score of all benchmarks in each domain. In each column, the highest score is **bolded** and the second-highest is underlined.

Superiority of Data Efficiency : Mix-GRM achieves these gains with substantially less data. While *FARE-8B* relies on massive scaling (≈ 2.5M samples) to reach 75.9, *Mix-GRM* attains a comparable 75.1 in the SFT stage using merely 9K samples. This finding highlights that optimizing CoT mechanisms yields a substantially higher training signal density, enabling data efficiency compared to brute-force dataset expansion.

Switching Amplification via RLVR : Mix CoT maximizes the efficacy of the RLVR stage, unlocking greater performance gains than unstructured CoT. RLVR boosts *Mix-GRM* by 4.3 (75.1 → 79.4), compared to a 3.6 gain for *Base-GRM* (73.3 → 76.9). Consequently, the performance gap over the *Base-GRM* widens from 1.8 (SFT) to 2.5 (RLVR), confirming that the aligned mechanism offers a more exploitable base for the RL. Furthermore, our subsequent analysis (Sec 5) reveals that these gains are fundamentally underpinned by an emergent polarization in mechanism allocation, where RLVR sharpens the model's reasoning style to match task-specific demands.

4.3 Mechanism Efficiency

Table 2 reveals that mechanism efficacy is strictly task-dependent. In the SFT stage, we observe a **distinct performance trade-off**: B-CoT improves Preference via lateral coverage but degrades Correctness (72.2 → 70.2), whereas D-CoT enhances deductive soundness but fails in Preference (68.2 → 65.9). These results indicate that simply extending CoT length does not guarantee universal gains; while principle expansion facilitates multi-dimensional evaluation, it offers no inherent advantage for deep reasoning. However, *Mix-GRM* overcomes these limitations through a **synergistic mutual enhancement**. By integrating orthogonal strengths, it not only surpasses the *Base-GRM*

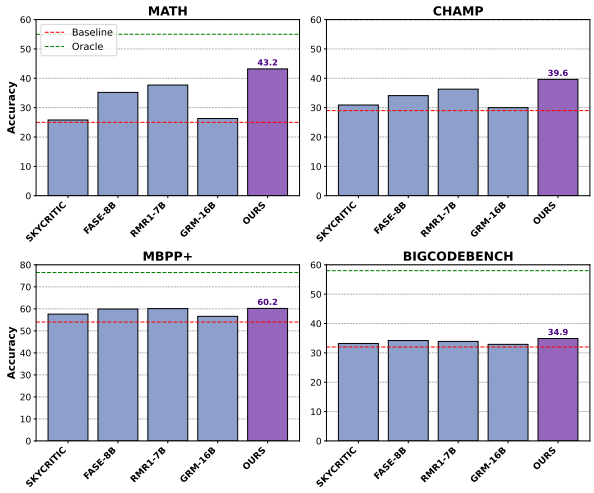

Figure 2: Best-of-10 performance across four challenging reasoning and coding benchmarks. Mix-GRM (ours) consistently achieves the highest accuracy across all tasks, effectively identifying solutions in both mathematical and code generation scenarios. Red and green lines denote random and oracle selection baselines.

(70.2 $\rightarrow$ 71.8) but surprisingly outperforms specialized single-mode models on their respective strongholds (*e.g.*, exceeding Depth-only on Correctness). This synergy becomes critical during the RLVR stage, where single-mode mechanisms encounter hard performance ceilings—most notably, *Mix-GRM (Breadth)* plateaus on correctness tasks. In contrast, the *Mix-GRM* enables RL optimization to reach a superior ceiling (78.7). This confirms that the **CoT structure itself acts as a bottleneck for RL optimization**; the mix structures does not merely inherit component strengths but constructs a robust reasoning framework that transcends the inherent limitations of isolated mechanisms.

4.4 Downstream Utility

To validate the practical utility of *Mix-GRM*, we apply it to two downstream applications: (i) serving as a reward signal for **Offline Reinforcement Learning**, and (ii) acting as a verifier for **Test-time Scaling**. We provide detailed descriptions of these application settings in Appendix C.

Reward Model for Offline Reinforcement Learning. In Offline RL via Direct Preference Optimization (DPO) (Rafailov et al., 2023), RM constructs high-quality preference pairs (y_w, y_l) to supervise policy alignment. Table 3 shows that models trained on these signals achieve a peak win rate of **12.1** in instruction-following, surpassing *FARE-8B* (12.0) and *RubricRM* (10.5). Crucially, this alignment gain does not compromise reasoning capabilities; in the math domain, Mix-GRM maintains a SOTA accuracy of **46.4**, edging out *RubricRM* (45.9) and *RM-R1-Instruct* (45.8). Specifically, Mix-GRM achieves 77.6% on GSM8K, demonstrating a clear lead over the SFT baseline (75.1%). These results confirm that Mix-GRM provides reliable supervision, enabling policies to internalize both helpfulness and correctness.

Reward Model for Test-time Scaling. For test-time scaling, leveraging increased inference-time compute to enhance generalization, Mix-GRM functions as a robust verifier to re-rank candidates to identify the optimal solution via Best-of-N selection. Following the JETTS protocol (Zhou et al., 2025b), we evaluate $N = 10$ samples from a Llama-3.1-8B generator across 4 diverse benchmarks: MATH and CHAMP (math), as well as MBPP+ and BigCodeBench (coding). As shown in Figure 2, our method consistently secures the highest accuracy, setting a new SOTA for 8B-scale rerankers. The performance advantage is particularly pronounced in reasoning-heavy tasks; for instance, on MATH, our model achieves an accuracy of 43.2%, outperforming the RL-driven *RM-R1* (37.7%) and the data-intensive *FARE-8B* (35.2%). This confirms that ours provides a more discriminative signal for logical verification than methods relying on massive data scaling or generic RL.

5 Analysis

Switching CoT Mechanism Analysis. Visualizing structural transformations (Figure 3) reveals how our pipeline reshapes reasoning mechanisms. Here, Single-mode strategies show extreme trade-offs: **Mix-GRM (Breadth)** expands horizontally (high principle count), while **Mix-GRM (Depth)** extends vertically (long judgments). In contrast, **Mix-GRM (SFT)** achieves a robust union of both, which is further expanded into a broader reasoning manifold by **RLVR**. First, the polarization of Breadth and Depth baselines confirms the rigidity of static templates, which create capability blind spots by sacrificing either reasoning depth or semantic coverage. Second, the balanced profile of **Mix-GRM (SFT)** indicates successful internalization of different distinct mechanisms. Most pivotally, the global expansion during **RLVR** validates our hypothesis of mechanism polarization. By optimizing for verdict accuracy, the model spontaneously converges on domain-specific mechanism biases—amplifying D-CoT for correctness while

reinforcing B-CoT for preference. This emergent specialization confirms that our proposed alignment is not a handcrafted heuristic, but an inherent structural necessity discovered by the model to maximize evaluation efficacy.

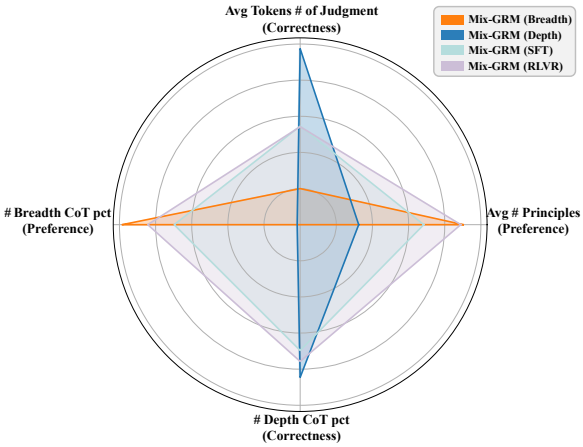

Figure 3: **Structural evolution of CoT mechanisms.** The chart tracks 4 indicators: the average token length per judgment, average principle count, and the percentage of CoT classified as having Breadth or Depth characteristics.

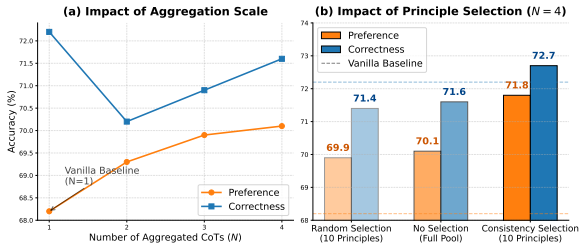

Figure 4: **Ablation of B-CoT synthesis.** (a) Aggregation Scale: Performance as aggregated rationales (N) increases from 1 (Vanilla) to 4. (b) Principle Selection: Comparison of Random, Full, and Consistency (Top-10) selection from the $N = 4$ pool. Orange/blue lines denote Preference/Correctness; dashed lines indicate the Vanilla baseline.

Scaling & Selection Analysis. To understand the mechanics of B-CoT, we decouple the impact of quantity (aggregation scale) from quality (principle selection) as shown in Figure 4.1) Quantity (Scaling): Figure 4(a) demonstrates that performance improves monotonically as the number of parallel CoTs (N) increases from 1 to 4. This confirms that "breadth" functions by expanding coverage; by aggregating diverse perspectives, the model minimizes the risk of overlooking critical error patterns.2) Quality (Selection): However, more is not always better. Figure 4(b) compares three strategies within the $N = 4$ pool:

CASE 1: PREFERENCE DOMAIN (B-COT WINS)
Inst (JP): アフガニスタン... (Is Afghanistan a puppet of Pakistan?) **Resp A (EN):** Detailed history... **Resp B (JP):** アフガニスタンがパキス...
✗ **Vanilla** "A provides comprehensive history." (Ignored language). *Verdict: [[A]].*
✗ **D-CoT** "Deep analysis of history..." (**Tunnel Vision**). ... *Verdict: [[A]].*
✓ **B-CoT** **Multi-dim Scan:** "1. **Principle of Linguistic Alignment:** "Assistant B's response is in Japanese... Sub-Verdict: «B»" ... *Verdict: [[B]].*
CASE 2: CORRECTNESS DOMAIN (D-COT WINS)
Inst: On the basis of oxidation-reduction potential,which of the following is most likely to occur? **Resp A:** The order is Alkali > ... > Zn > ... > Ag, ... **D. Zn + ...** (Correct) **Resp B:** Alkali > Alkaline earth ... **B. Mg+K...** (Logic Error: Mg>K)
✗ **Vanilla** **Surface Length:** "B analyzes more options and is longer." *Verdict: [[B]].*
✗ **B-CoT** **Superficial Heuristic:** "B covers options A-H comprehensively." (No verification). *Verdict: [[B]].*
✓ **D-CoT** **Rigorous Analysis:** "Check B: Claims Mg displaces K (False, K>Mg). Check D: Valid. Pick A." *Verdict: [[A]].*

Table 4: **Simplified Case Study**. **Case 1**: B-CoT catches language mismatch. **Case 2**: D-CoT verifies logical steps. The detailed Case is shown in Table. 5

$\text{Breadth}_{\text{Rand}}$, $\text{Breadth}_{\text{Full}}$, and $\text{Breadth}_{\text{Top10}}$, where Top-10 means ten most frequent principles appearing across 4 CoTs. We observe a clear hierarchy: $\text{Breadth}_{\text{Top10}} > \text{Breadth}_{\text{Full}} > \text{Breadth}_{\text{Rand}}$. While the full pool improves over random sampling, it is the Top-10 consensus that achieves the highest gains ($71.8/72.7$). This suggests a denoising effect where low-frequency principles introduce noise, while high-frequency ones form a more robust "reasoning consensus." Thus, representativeness—not just volume—is vital for robust breadth.

Case Study. Table 4 elucidates the structural drivers of the observed trade-off. In preference, B-CoT acts as a multi-dimensional scanner, identifying lateral mismatches (*e.g.*, tone) that D-CoT misses due to attentional tunneling. Conversely, for correctness, D-CoT functions as a probe, exposing factual hallucinations (*e.g.*, $K > Mg$) that B-CoT overlooks by mistaking superficial formatting for logical validity. This confirms that while Breadth ensures multi-faceted alignment, Depth remains the non-negotiable driver for rigorous evaluation.

6 Conclusion

This work demonstrates that beyond mere length scaling, the reliability of GRMs is fundamentally driven by the integration of different reasoning mechanisms. By introducing Mix-GRM, we prove that the frontier of reward modeling lies in synergizing two orthogonal reasoning mechanisms: B-CoT for multi-dimensional coverage and D-CoT

for judgment soundness. Through mechanism-adaptive alignment, Mix-GRM ensures that the RM's reasoning mechanism is precisely calibrated to the nature of the task. Ultimately, these findings shift the focus of GRM development from brute-force expansion to structural optimization.

Limitations

While Mix-GRM significantly enhances evaluation reliability through mechanism alignment, we identify two primary limitations that warrant further investigation:

Granularity of the Reasoning Manifold. Our framework successfully captures the double dissociation between Subjective Preference and Objective Correctness, which we identify as the dominant axes of the reasoning manifold. However, this dichotomy represents a coarse-grained mapping of the diverse alignment landscape. Real-world tasks often exist on a continuous spectrum or involve hybrid demands that intricately blend deductive rigor with multi-dimensional nuances. While we prove that the model's reasoning structure spontaneously converges toward these two primary poles, our current categorization may act as a low-rank approximation of a higher-dimensional space of mechanisms. Future work could explore more granular taxonomies to achieve even more precise task-mechanism calibration.

Rigidity in Ambiguous Task Boundaries. Our analysis demonstrates that RLVR induces an intrinsic convergence toward specialized reasoning poles. However, this emergent polarization may introduce a degree of structural rigidity when encountering hybrid tasks that do not fit neatly into the "Subjective vs. Objective" dichotomy. For instance, tasks that require both factual precision and sophisticated stylistic nuance may demand a dynamic fusion of B-CoT and D-CoT. While our current framework focuses on aligning specialized mechanisms with their respective domains, the spontaneous sharpening of reasoning styles might come at the cost of generalist flexibility in highly nuanced, cross-domain scenarios. Future research could explore adaptive, soft-routing mechanisms that allow for a more fluid transition across the reasoning manifold.

References

Zachary Ankner, Mansheej Paul, Brandon Cui, Jonathan D. Chang, and Prithviraj Ammanabrolu. 2024. Critique-out-loud reward models. *Preprint*, arXiv:2408.11791.

Maciej Besta, Florim Memedi, Zhenyu Zhang, Robert Gerstenberger, Guangyuan Piao, Nils Blach, Piotr Nyczyk, Marcin Copik, Grzegorz Kwaśniewski, Jürgen Müller, Lukas Gianinazzi, Ales Kubicek, Hubert Niewiadomski, Aidan O'Mahony, Onur Mutlu, and Torsten Hoefler. 2025. Demystifying chains, trees, and graphs of thoughts. *Transactions on Pattern Analysis and Machine Intelligence*, page 10967–10989.

Nuo Chen, Zhiyuan Hu, Qingyun Zou, Jiaying Wu, Qian Wang, Bryan Hooi, and Bingsheng He. 2025a. Judgelrm: Large reasoning models as a judge. *Preprint*, arXiv:2504.00050.

Xiusi Chen, Gaotang Li, Ziqi Wang, Bowen Jin, Cheng Qian, Yu Wang, Hongru Wang, Yu Zhang, Denghui Zhang, Tong Zhang, Hanghang Tong, and Heng Ji. 2025b. RM-R1: Reward modeling as reasoning. *Preprint*, arXiv:2505.02387.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. 2024. UltraFeedback: Boosting language models with scaled ai feedback. *Preprint*, arXiv:2310.01377.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. In *Conference on Empirical Methods in Natural Language Processing*.

Yann Dubois, Percy Liang, and Tatsunori Hashimoto. 2024. Length-controlled alpacaeval: A simple debiasing of automatic evaluators. In *Conference on Language Modeling*.

Evan Frick, Tianle Li, Connor Chen, Wei-Lin Chiang, Anastasios Nikolas Angelopoulos, Jiantao Jiao, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. 2025. How to evaluate reward models for RLHF. In *The Thirteenth International Conference on Learning Representations*.

Anisha Gunjal, Anthony Wang, Elaine Lau, Vaskar Nath, Yunzhong He, Bing Liu, and Sean Hendryx. 2025. Rubrics as rewards: Reinforcement learning beyond verifiable domains. *Preprint*, arXiv:2507.17746.

Seungju Han, Kavel Rao, Allyson Ettinger, Liwei Jiang, Bill Yuchen Lin, Nathan Lambert, Yejin Choi, and Nouha Dziri. 2024. WILDGUARD: open one-stop

moderation tools for safety risks, jailbreaks, and refusals of llms. In *International Conference on Neural Information Processing Systems*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the MATH dataset. In *Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Subbarao Kambhampati, Karthik Valmeekam, Lin Guan, Mudit Verma, Kaya Stechly, Siddhant Bhambri, Lucas Saldyt, and Anil Murthy. 2024. Llms can't plan, but can help planning in llm-modulo frameworks. *Preprint*, arXiv:2402.01817.

Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, and Minjoon Seo. 2024. Prometheus: Inducing fine-grained evaluation capability in language models. In *International Conference on Learning Representations*.

Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. 2016. MAWPS: A math word problem repository. In *Conference of the North American Chapter of the Association for Computational Linguistics*, pages 1152–1157.

Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, Noah A. Smith, and Hannaneh Hajishirzi. 2024. RewardBench: Evaluating reward models for language modeling. *Preprint*, arXiv:2403.13787.

Hung Le, Yue Wang, Akhilesh Deepak Gotmare, Silvio Savarese, and Steven C.H. Hoi. 2022. CodeRL: mastering code generation through pretrained models and deep reinforcement learning. In *International Conference on Neural Information Processing Systems*.

Harrison Lee, Samrat Phatale, Hassan Mansoor, Thomas Mesnard, Johan Ferret, Kellie Lu, Colton Bishop, Ethan Hall, Victor Carbune, Abhinav Rastogi, and Sushant Prakash. 2024. Rlaif vs. rlhf: scaling reinforcement learning from human feedback with ai feedback. In *International Conference on Machine Learning*.

Tianle Li, Wei-Lin Chiang, Evan Frick, Lisa Dunlap, Banghua Zhu, Joseph E. Gonzalez, and Ion Stoica. 2024. From live data to high-quality benchmarks: The arena-hard pipeline.

Yi-Chen Li, Tian Xu, Yang Yu, Xuqin Zhang, Xiong-Hui Chen, Zhongxiang Ling, Ningjing Chao, Lei Yuan, and Zhi-Hua Zhou. 2025a. Generalist reward models: Found inside large language models. *Preprint*, arXiv:2506.23235.

Zhong-Zhi Li, Duzhen Zhang, Ming-Liang Zhang, Jiaxin Zhang, Zengyan Liu, Yuxuan Yao, Haotian Xu, Junhao Zheng, Pei-Jie Wang, Xiuyi Chen, Yingying Zhang, Fei Yin, Jiahua Dong, Zhiwei Li, Bao-Long Bi, Ling-Rui Mei, Junfeng Fang, Xiao Liang, Zhijiang Guo, and 2 others. 2025b. From system 1 to system 2: A survey of reasoning large language models. *Preprint*, arXiv:2502.17419.

Zhuang Li, Yuncheng Hua, Thuy-Trang Vu, Haolan Zhan, Lizhen Qu, and Gholamreza Haffari. 2025c. SCAR: Data selection via style consistency-aware response ranking for efficient instruction-tuning of large language models. *Preprint*, arXiv:2406.10882.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *Preprint*, arXiv:2305.20050.

Chengwu Liu, Ye Yuan, Yichun Yin, Yan Xu, Xin Xu, Zaoyu Chen, Yasheng Wang, Lifeng Shang, Qun Liu, and Ming Zhang. 2025a. Safe: Enhancing mathematical reasoning in large language models via retrospective step-aware formal verification. In *Annual Meeting of the Association for Computational Linguistics*.

Chris Yuhao Liu, Liang Zeng, Jiacai Liu, Rui Yan, Jujie He, Chaojie Wang, Shuicheng Yan, Yang Liu, and Yahui Zhou. 2024. Skywork-reward: Bag of tricks for reward modeling in llms. *Preprint*, arXiv:2410.18451.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and LINGMING ZHANG. 2023. Is your code generated by chatGPT really correct? rigorous evaluation of large language models for code generation. In *Conference on Neural Information Processing Systems*.

Tianci Liu, Ran Xu, Tony Yu, Ilgee Hong, Carl Yang, Tuo Zhao, and Haoyu Wang. 2025b. OpenRubrics: Towards scalable synthetic rubric generation for reward modeling and llm alignment. *Preprint*, arXiv:2510.07743.

Yantao Liu, Zijun Yao, Rui Min, Yixin Cao, Lei Hou, and Juanzi Li. 2025c. RM-bench: Benchmarking reward models of language models with subtlety and style. In *International Conference on Learning Representations*.

Zijun Liu, Peiyi Wang, Runxin Xu, Shirong Ma, Chong Ruan, Peng Li, Yang Liu, and Yu Wu. 2025d. Inference-time scaling for generalist reward modeling. *Preprint*, arXiv:2504.02495.

Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark, and Ashwin Kalyan. 2023. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. *Preprint*, arXiv:2209.14610.

Saumya Malik, Valentina Pyatkin, Sander Land, Jacob Morrison, Noah A. Smith, Hannaneh Hajishirzi, and Nathan Lambert. 2025. RewardBench

2: Advancing reward model evaluation. *Preprint*, arXiv:2506.01937.

Yujun Mao, Yoon Kim, and Yilun Zhou. 2024. CHAMP: A competition-level dataset for fine-grained analyses of llms' mathematical reasoning capabilities. *Preprint*, arXiv:2401.06961.

Xuefei Ning, Zinan Lin, Zixuan Zhou, Zifu Wang, Huazhong Yang, and Yu Wang. 2024. Skeleton-of-thought: Prompting LLMs for efficient parallel generation. In *International Conference on Learning Representations*.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *International Conference on Neural Information Processing Systems*.

Jiayi Pan, Xiuyu Li, Long Lian, Charlie Snell, Yifei Zhou, Adam Yala, Trevor Darrell, Kurt Keutzer, and Alane Suhr. 2025. Learning adaptive parallel reasoning with language models. *Preprint*, arXiv:2504.15466.

Junsoo Park, Seungyeon Jwa, Ren Meiying, Daeyoung Kim, and Sanghyuk Choi. 2024. OffsetBias: Leveraging debiased data for tuning evaluators. In *Findings of the Association for Computational Linguistics: EMNLP*, pages 1043–1067.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. In *Conference on Neural Information Processing Systems*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. DeepSeekMath: Pushing the limits of mathematical reasoning in open language models. *Preprint*, arXiv:2402.03300.

Wei Shen, Guanlin Liu, YuYue, Ruofei Zhu, Qingping Yang, Chao Xin, and Lin Yan. 2025. Exploring data scaling trends and effects in reinforcement learning from human feedback. In *Annual Conference on Neural Information Processing Systems*.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Preprint*, arXiv:2303.11366.

Zayne Rea Sprague, Fangcong Yin, Juan Diego Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. 2025. To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning. In *International Conference on Learning Representations*.

DeepSeek-AI Team. 2025a. DeepSeek-R1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

Kimi Team. 2025b. Kimi k1.5: Scaling reinforcement learning with llms. *Preprint*, arXiv:2501.12599.

Qwen Team. 2025c. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

Vijay Viswanathan, Yanchao Sun, Shuang Ma, Xiang Kong, Meng Cao, Graham Neubig, and Tongshuang Wu. 2025. Checklists are better than reward models for aligning language models. *Preprint*, arXiv:2507.18624.

Junqiao Wang, Zeng Zhang, Yangfan He, Zihao Zhang, Xinyuan Song, Yuyang Song, Tianyu Shi, Yuchen Li, Hengyuan Xu, Kunyu Wu, Xin Yi, Zhongwei Wan, Xinhang Yuan, Zijun Wang, Kuan Lu, Menghao Huo, Tang Jingqun, Guangwu Qian, Keqin Li, and 2 others. 2025a. Enhancing code llms with reinforcement learning in code generation: A survey. *Preprint*, arXiv:2412.20367.

Peiyi Wang, Lei Li, Zhihong Shao, Runxin Xu, Damai Dai, Yifei Li, Deli Chen, Yu Wu, and Zhifang Sui. 2024a. Math-shepherd: Verify and reinforce LLMs step-by-step without human annotations. In *Annual Meeting of the Association for Computational Linguistics*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. *Preprint*, arXiv:2203.11171.

Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. 2024b. Mmlu-pro: a more robust and challenging multi-task language understanding benchmark. In *International Conference on Neural Information Processing Systems*.

Zhilin Wang, Jiaqi Zeng, Olivier Delalleau, Hoo-Chang Shin, Felipe Soares, Alexander Bukharin, Ellie Evans, Yi Dong, and Oleksii Kuchaiev. 2025b. Helpsteer3-preference: Open human-annotated preference data across diverse tasks and languages. *Preprint*, arXiv:2505.11475.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Chenxi Whitehouse, Tianlu Wang, Ping Yu, Xian Li, Jason Weston, Ilia Kulikov, and Swarnadeep Saha. 2025. J1: Incentivizing thinking in llm-as-a-judge via reinforcement learning. *Preprint*, arXiv:2505.10320.

Austin Xu, Xuan-Phi Nguyen, Yilun Zhou, Chien-Sheng Wu, Caiming Xiong, and Shafiq Joty. 2025. Foundational automatic evaluators: Scaling multi-task generative evaluator training for reasoning-centric domains. *Preprint*, arXiv:2510.17793.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of thoughts: deliberate problem solving with large language models. In *International Conference on Neural Information Processing Systems*.

Edward Yeo, Yuxuan Tong, Morry Niu, Graham Neubig, and Xiang Yue. 2025. Demystifying long chain-of-thought reasoning in llms. *Preprint*, arXiv:2502.03373.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng YU, Zhengying Liu, Yu Zhang, James Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2024. Metamath: Bootstrap your own mathematical questions for large language models. In *International Conference on Learning Representations*.

Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Xian Li, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. 2024. Self-rewarding language models. In *International Conference on Machine Learning*.

Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. 2025a. Generative verifiers: Reward modeling as next-token prediction. In *International Conference on Learning Representations*.

Qiyuan Zhang, Fuyuan Lyu, Zexu Sun, Lei Wang, Weixu Zhang, Wenyue Hua, Haolun Wu, Zhihan Guo, Yufei Wang, Niklas Muennighoff, Irwin King, Xue Liu, and Chen Ma. 2025b. A survey on test-time scaling in large language models: What, how, where, and how well? *Preprint*, arXiv:2503.24235.

Qiyuan Zhang, Yufei Wang, Yuxin Jiang, Liangyou Li, Chuhan Wu, Yasheng Wang, Xin Jiang, Lifeng Shang, Ruiming Tang, Fuyuan Lyu, and Chen Ma. 2025c. Crowd comparative reasoning: Unlocking comprehensive evaluations for LLM-as-a-judge. In *Annual Meeting of the Association for Computational Linguistics*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Tong Zheng, Hongming Zhang, Wenhao Yu, Xiaoyang Wang, Runpeng Dai, Rui Liu, Huiwen Bao, Chengsong Huang, Heng Huang, and Dong Yu. 2025. Parallel-R1: Towards parallel thinking via reinforcement learning. *Preprint*, arXiv:2509.07980.

Enyu Zhou, Guodong Zheng, Binghai Wang, Zhiheng Xi, Shihan Dou, Rong Bao, Wei Shen, Limao Xiong, Jessica Fan, Yurong Mou, Rui Zheng, Tao Gui, Qi Zhang, and Xuanjing Huang. 2025a. RMB: Comprehensively benchmarking reward models in LLM alignment. In *International Conference on Learning Representations*.

Yilun Zhou, Austin Xu, Peifeng Wang, Caiming Xiong, and Shafiq Joty. 2025b. Evaluating judges as evaluators: The jetts benchmark of llm-as-judges as test-time scaling evaluators. *Preprint*, arXiv:2504.15253.

Terry Yue Zhuo, Vu Minh Chien, Jenny Chim, Han Hu, Wenhao Yu, Ratnadira Widyasari, Imam Nur Bani Yusuf, Haolan Zhan, Junda He, Indraneil Paul, Simon Brunner, Chen GONG, James Hoang, Armel Randy Zebaze, Xiaoheng Hong, Wen-Ding Li, Jean Kaddour, Ming Xu, Zhihan Zhang, and 14 others. 2025. Bigcodebench: Benchmarking code generation with diverse function calls and complex instructions. In *International Conference on Learning Representations*.

Table 5: Case Study. **Case 1** shows how Breadth-CoT aggregates diverse principles to identify subtle preference nuances. **Case 2** shows how Depth-CoT performs step-by-step verification to catch logical errors.

CASE 1: PREFERENCE DOMAIN (BREADTH-COT WINS)	
Instruction: アフガニスタンがパキスタンの傀儡というのは本当ですか？ (Is it true that Afghanistan is a puppet of Pakistan?)	
Response A (Rejected):	**Response B (Chosen):**
[Language: English] "A sensitive topic! ...While Pakistan has historically exerted significant influence..."	**[Language: Japanese]**「アフガニスタンとパキスタンの関係については...簡単に『傀儡』と断言するのは適切ではなく...」
Reasoning Comparison	
✗ Vanilla-CoT	Assistant A offers a comprehensive breakdown of historical context... **Verdict: [[A]]** (Fail: Ignored language mismatch)
✓ **Breadth-CoT**	1. **Principle of Linguistic Alignment:** "Assistant B's response is in Japanese... Sub-Verdict: «B»" 2. **Principle of Contextual Nuance:** "Assistant A provides a detailed explanation... Sub-Verdict: «A»" 3. **Principle of Cultural Sensitivity:** "..." **Final Verdict: [[B]]** (Success)
✗ Depth-CoT	1. **Principle of Comprehensive Contextualization:** The evaluation needs deep reasoning, and let's solve the instruction: アフガニスタンが'.... Based on this, Assistant A excels... Sub-Verdict: «A» 2. **Principle of Balanced Presentation:** ... **Final Verdict: [[A]]** (Fail: Tunnel vision on content depth)

CASE 2: CORRECTNESS DOMAIN (DEPTH-COT WINS)	
Instruction: On the basis of oxidation-reduction potential, which of the following is most likely to occur? ...	
Response A (Chosen):	**Response B (Rejected):**
"The reactivity of metals... The order is: Alkali > ... > Zn > ... > Ag. Based on this, the most likely reaction is: **D. Zn + 2Ag(NO3)...**" (Correct)	"The reactivity... Alkali > Alkaline earth... Analysis: **B. Mg + 2KNO3 -> 2K + Mg(NO3)2**." (**Error:** K > Mg.)
Reasoning Comparison	
✗ Vanilla-CoT	To evaluate the responses, let's consider factors: 1. Helpfulness; 2. Relevance... Assistant B goes further by analyzing multiple options... **Final Verdict: [[B]]** (Fail: Fooled by length/detail)
✗ Breadth-CoT	1. **Principle of Comprehensive Option Analysis:** "Assistant B analyzes all provided options (A-H)... Sub-Verdict: «B»" 2. **Principle of Informative Detail:** "Assistant B includes more elements... Sub-Verdict: «B»" **Final Verdict: [[B]]** (Fail: Superficial heuristic)
✓ **Depth-CoT**	1. **Principle of Comprehensive Analysis:** The evaluation needs to deep reasoning, and let's solve the instruction: To determine the most likely reaction... The correct order is Alkali > ... Given the options, the most likely reaction is Option D. ... Assistant B correctly identifies the importance but incorrectly identifies Option B... Sub-Verdict: «A» 2. **Principle of Direct Relevance:** Assistant A directly addresses the question... Sub-Verdict: «A» **Final Verdict: [[A]]** (Success)

A Case Study

Table 5 elucidates the structural mechanisms behind the observed double dissociation. Case 1 demonstrates why Breadth-CoT dominates preference tasks: acting as a multi-dimensional scanner, it successfully penalizes a detailed but language-mismatched response by validating lateral constraints (*e.g.*, *Linguistic Alignment*), whereas Depth-CoT exhibits attentional tunneling," fixating on verifying historical facts while missing the high-level language mismatch. Conversely, Case 2 reveals why *Depth-CoT* is essential for correctness: its step-by-step derivation acts as a logic probe, allowing it to spot subtle factual hallucinations (*e.g.*, $K > Mg$) hidden within a lengthy explanation. Here, *Breadth-CoT* actively fails due to feature interference," where it mistakes superficial comprehensiveness (length and formatting) for logical validity. This confirms that while Breadth is necessary for satisfying diverse user preferences, Depth is the non-negotiable driver for rigorous verification.

B Training Implementation

B.1 Hyperparameters Setting

We provide the detailed hyperparameter settings in the Table 6 and Table 7.

Hyperparameters	Values
Epochs	2
Learning rate	2e−5
Batch Size	128 (gradient accumulation steps = 16)
Seq Length	12, 288
Weight Decay	0.
Warmup	5% linear warmup

Table 6: Hyperparameter settings for SFT.

Hyperparameters	Values
Training Steps	100
Learning Rate	1e−6
Batch Size	128
KL Loss Coefficient	0.001
KL Coefficient	0.001
Rollouts	n= 8 using vLLm with temperature 0.8

Table 7: Hyperparameter settings for RL.

B.2 Training Data Source Details

To cultivate general rewarding capabilities, it is essential to curate a training corpus that encompasses diversified real-world scenarios. We construct our dataset by performing stratified random sampling from representative data sources, ensuring balanced coverage across distinct alignment domains, including general chat, STEM, coding, math, safety, multilingual, and instruction following. The specific source datasets, their corresponding domains, and the sampling statistics are detailed in Table 8.

Source Dataset	Domain	Samples
`HelpSteer-3` (Single-Turn)	General Chat	4,973
	STEM	2,321
	Code	4,322
	Multilingual	3,260
`Code-Preference`	Code	4,000
`Math-DPO`	Math	4,000
`WildGuard`	Safety	4,000
`OffsetBias`	Instruction Following	4,000
Total	–	**30,876**

Table 8: Composition and statistics of the training data sampled from domain-specific sources.

B.3 Training Data Synthesis Details

To synthesize the CoT data for SFT, we utilized `DeepSeek-v3` (0324 snapshot) as the backbone generator. The generation process was configured with a sampling temperature of $T = 0.8$ to promote diversity in the trajectories while maintaining logical coherence. Notably, we abstain from consistency filtering: Contrary to common practices that discard samples where the synthesized verdict diverges from the ground-truth human label, our empirical verification reveals that training on the full synthesized CoTs yields superior performance compared to aggressive filtering, regardless of verdict consistency.

B.4 Training Offline Reinforcement Learning Details

To strictly control for temporal data leakage and ensure a fair comparison with the release dates of our evaluation benchmarks, we select **Llama-3-8B** as our base foundation model. The offline reinforcement learning pipeline consists of two phases: SFT initialization and DPO.

Policy Initialization (SFT). We first derive a supervised policy model by fine-tuning Llama-3-8B on a composite dataset. This dataset ensures basic instruction-following and reasoning capabilities, consisting of the **UltraChat** dataset (Ding et al., 2023) and a random subset of 40K samples from **MetaMathQA** (Yu et al., 2024). We train the model for 2 epochs using a learning rate of $2e{-}5$ and a maximum sequence length of $2,048$ tokens. This SFT model serves as the initial policy π_{ref} for the subsequent DPO stage.

Table 9: Task coverage of the evaluated general reward benchmarks.

Benchmark	Tasks	Samples
RewardBench	Chat, Math, Code, Safety	2,985
RewardBench-v2	Focus, IF, Factuality, Math, Safety, Ties	1,865
RM-Bench	Chat, Math, Code, Safety	11,943
RMB	Harmfulness, Helpfulness (General, Code)	14,725
PPE (Exclude Tie)	Chat, MMLU-Pro, GPQA, IFEval, MBPP	22,991

DPO Data Construction via RM Labeling. To evaluate the practical utility of different RMs, we employ them to annotate preferences on a unified source dataset. The prompt source comprises 10K instructions randomly sampled from **UltraFeedback** (Cui et al., 2024) and 40K instructions from **MetaMathQA**. For each instruction x, we generate $N = 5$ diverse candidate responses using `gpt-4o-mini` with a temperature of 0.8.

We adopt a **Pairwise Scoring Aggregation** strategy to construct the final preference pairs (x, y_w, y_l). Specifically, for the set of 5 responses, we generate all possible combinations of pairs ($\binom{5}{2} = 10$ pairs). The target RM evaluates each pair, assigning $+1$ point to the preferred response (chosen) and 0 to the non-preferred one (rejected). After traversing all pairs, we calculate the cumulative score for each response. The response with the highest total score is selected as the positive sample (y_w), and the response with the lowest total score is selected as the negative sample (y_l). These labeled pairs are then used to train the policy via DPO.

C Evaluation Implementation

C.1 Core Benchmarks

There is a list of benchmarks and corresponding task coverage.

C.2 Benchmarks for Offline Reinforcement Learning Evaluation

To comprehensively assess the policy derived from DPO, we conduct evaluations across two distinct

domains: mathematical reasoning and open-ended instruction following.

Mathematical Reasoning. We employ a suite of four challenging datasets to evaluate the model's deductive logic and problem-solving capabilities: GSM8k (Cobbe et al., 2021), MATH (Hendrycks et al., 2021), MAWPS (Koncel-Kedziorski et al., 2016), and TabMWP (Lu et al., 2023). These benchmarks cover a wide spectrum of difficulty, ranging from grade-school arithmetic to competition-level mathematics and tabular processing.

Instruction Following. For general alignment and conversational versatility, we utilize two widely adopted benchmarks: AlpacaEval-2 (Dubois et al., 2024) and Arena-Hard v0.1 (Li et al., 2024). Evaluation is performed using an auto-evaluator in a head-to-head setting, where the model's responses are compared against a baseline reference to determine win rates. We strictly adhere to the officially recommended configurations for reproducibility.

C.3 Benchmarks for Test-time Scaling Evaluation

Following the **JETTS** setup (Zhou et al., 2025b), we perform Best-of-10 reranking evaluations where the model selects the optimal solution from a mixed pool of candidate responses. We report results on the four most challenging subsets of the benchmark: **MATH** (Hendrycks et al., 2021) for mathematical reasoning, **CHAMP** (Mao et al., 2024) for competition-level math, along with **MBPP+** (Liu et al., 2023) and **BigCodeBench** (Zhuo et al., 2025) for code generation. This selection tests the model's ability to identify correct reasoning paths in complex scenarios.

D Prompts Template

To align with established community standards, our Vanilla-CoT generation employs the representative prompts originally introduced in MT-Bench (Zheng et al., 2023) and RewardBench (Lambert et al., 2024). Upon generating the raw Vanilla-CoT using the standard prompts, we employ a specialized extraction prompt to parse the unstructured text into the modular "Principle–Judgment–Verdict" schema. Leveraging the parsed schemas, we introduce specialized prompts to synthesize the two target morphologies. For Breadth-CoT, the synthesis process entails merging modular components derived from at least two Vanilla-CoT responses, followed by a deduplication step to ensure diverse coverage. In contrast, the synthesis of Depth-CoT relies on a reasoning-guided evaluation mechanism. We initially prompt the model to reason the instruction deeply. We then use this generated reasoning to ground the re-assessment of selected principles extracted from the parsed schemas, discarding their previous rationales to ensure the new judgments are purely driven by rigorous reasoning.

E Reward Model Performance Across Preference and Correctness

To provide a more granular view of our model's efficacy, we report detailed performance across specific tasks based on the meta-data provided by each benchmark. We categorize these tasks into two distinct tables: Table 10 for subjective preference tasks and Table 11 for objective correctness tasks. This fine-grained reporting serves as a detailed decomposition of the mechanism-level performance discussed in the main text, offering deeper empirical evidence for the mechanism-task synergy between B-CoT and D-CoT.

F Sensitivity and Robustness

The data synthesis process consists of two primary phases: the sampling of raw rationales and the subsequent synthesis of B-CoT/D-CoT. To guarantee the reliability of this pipeline, we conduct empirical analyses focusing on pipeline stability and robustness to noise.

Quantitative Evaluation of Pipeline Stability. To ensure our extraction and refinement modules do not introduce systematic errors or degrade data quality, we tracked the solution accuracy of the synthesized data at each pipeline stage on the training set. As shown in Table 12, the accuracy remains highly stable across the transformations. The merge step (B-CoT) resolves contradictions and improves accuracy, while the D-CoT generation maintains high fidelity to the correct reasoning paths. This confirms that the intermediate processing steps reliably preserve data quality without catastrophic degradation.

Robustness to Noise. A common concern with LLM-generated rationales is the necessity of strict filtering to ensure perfect correctness. We conducted an ablation study during the SFT stage to measure the impact of noise. We compared a model

Prompt for Vanilla-CoT Generation

```
Please act as an impartial judge and evaluate the quality of the responses provided by two AI
assistants to the user question displayed below. You should choose the assistant that follows
the user's instructions and answers the user's question better. Your evaluation should consider
as many factors as possible. Begin your evaluation by comparing the two responses and provide a
through reasoning. Avoid any position biases and ensure that the order in which the responses were
presented does not influence your decision. Do not allow the length of the responses to influence
your evaluation. Do not favor certain names of the assistants. Be as objective as possible. After
providing your reasoning, output your final verdict by strictly following this format: [[A]]if
assistant A is better, [[B]]if assistant B is better.
[Instruction]
instruction
[The Start of Assistant A's Answer]
{response_a}
[The End of Assistant A's Answer]

[The Start of Assistant B's Answer]
{response_b}
[The End of Assistant B's Answer]
```

trained on 9K strictly verified CoT data (filtered for correct final verdicts) against one trained on 9K noisy CoT data (without strict correctness filtering). As shown in Table 13, the SFT performance difference is negligible. This indicates that our method is highly robust to noise in the synthesized rationales and does not require perfect accuracy from the extraction pipeline to be effective. Consequently, we adopt the unfiltered data setting for our main experiments to significantly reduce the computational cost of data curation without sacrificing performance.

Furthermore, we utilize the open-weights DEEPSEEK-V3 model for both schema extraction and raw rationale generation. The effectiveness of our synthesis pipeline with an open-source model further underscores its robustness and generalizability, proving it is not overly sensitive to the choice of the underlying LLM.

G Emergent Polarization Driven by RLVR

A critical aspect of our method is the emergent polarization toward domain-specific reasoning mechanisms. A natural question arises: whether the RLVR stage genuinely discovers this polarization, or merely reinforces the priors established during SFT, given that the SFT training data explicitly pairs specific domains with distinct CoT formats (*i.e.*, B-CoT for Preference and D-CoT for Correctness).

To investigate the impact of RLVR on the CoT structure, we systematically analyzed the distribution of generated CoT structures on the test set. We established specific structural indicators to classify the outputs and verify their alignment with the target domains. Specifically, rationales containing more than four distinct principles were classified as B-CoT, while those containing the phrase "The evaluation needs to deep reasoning" were classified as D-CoT.

As presented in Table 14, although our 9K SFT training samples were strictly aligned (100%) with their respective domains, the model following the SFT stage only achieved a 73% structural match rate during inference. However, after applying RLVR, which relies exclusively on final verdict supervision and provides zero explicit structural labels, the structural alignment on the test set surged to 95%.

This substantial improvement demonstrates that the model does not merely inherit static priors from SFT. Instead, it learns to autonomously select and optimize the reasoning structure best suited for each domain during the reinforcement learning process. This confirms that the emergent polarization is effectively and actively driven by RLVR.

G.1 Computational Overhead and Token Cost Analysis

We quantify the computational overhead. We provide a detailed accounting of the compute and token costs incurred during training and inference to ensure a transparent comparison.

Algorithmic Fairness in Data Synthesis. We carefully designed our synthesis pipelines to en-

Table 10: Performance of RMs on preference-related sub-tasks. "Avg." is the average score among sub-tasks. Best per column is **bolded**; second-best is underlined.

Models	Reward Bench	Reward Bench-v2		RM Bench	RMB	PPE		Avg.
	CHAT	FOCUS	IF	CHAT	HELPFULNESS	HUMAN	IF	
Open-sourced Reward Models								
JudgeLRM-7B	75.8	46.8	31.3	68.4	79.3	60.2	54.3	59.4
RM-R1-7B (Distill)	75.3	58.7	24.4	58.7	63.2	57.1	53.0	55.8
RM-R1-7B (Instruct)	80.5	85.3	28.8	64.7	65.1	57.1	53.0	62.1
FARE-8B	85.0	78.4	36.3	66.9	**82.9**	63.4	55.7	67.0
RubricRM-8B	82.4	78.2	33.8	62.2	77.5	63.8	**66.0**	66.3
DeepSeek-GRM-16B	80.6	79.2	40.0	64.0	76.8	61.7	57.9	65.7
Our Proposed Reward Models								
SFT-trained								
Base-GRM	81.6	76.6	34.4	63.3	80.5	64.3	55.9	65.2
Mix-GRM (Breadth)	83.7	84.5	33.8	65.9	77.9	63.5	55.6	66.4
Mix-GRM (Depth)	80.3	72.2	28.1	70.6	70.1	63.1	54.1	62.6
Mix-GRM	84.9	79.6	31.9	71.2	78.7	62.0	56.3	66.4
RLVR-trained								
Base-GRM	83.0	86.7	29.4	68.5	73.8	65.8	57.1	66.3
Mix-GRM (Breadth)	**86.2**	88.9	28.8	70.1	79.2	**66.0**	55.3	67.8
Mix-GRM (Depth)	85.3	89.3	26.3	**75.6**	75.4	**66.0**	56.4	67.8
Mix-GRM	**86.2**	**91.3**	**37.5**	72.7	78.1	65.9	57.4	**69.9**

sure that the computational overhead for B-CoT and D-CoT remains strictly comparable. Specifically, B-CoT involves sampling initial rationales followed by a deterministic merge and deduplication step, which introduces no additional reasoning tokens. Conversely, D-CoT utilizes a two-step generation process: first generating a solve trace, and then synthesizing the final D-CoT. Consequently, both methods utilize approximately two reasoning passes. This parity ensures that our comparative analysis focuses purely on the structural efficacy of the reasoning rather than being confounded by raw compute disparities.

Token Cost Analysis. Table 15 details the average token counts per sample during Data Synthesis (SFT target tokens), RLVR Training (rollout tokens), and Inference.

As shown in Table 15, token consumption is highly comparable across D-CoT, B-CoT, and Mix-CoT. The difference in RLVR rollout lengths is marginal, confirming that all three configurations operate within the same order of magnitude of compute cost.

Table 11: Performance of RMs on correctness-related sub-tasks. "Avg." is the average within this block. Best per column is **bolded**; second-best is underlined.

Models	RewardBench		RewardBench-v2		RM-Bench		RMB	PPE				Avg.
	CODE	MATH	FACTUALITY	MATH	CODE	MATH	CODE	MMLU-PRO	MATH	GPQA	MBPP	
Open-sourced Reward Models												
JudgeLRM-7B	81.6	77.2	53.8	76.5	51.0	**86.7**	82.1	57.2	65.5	51.3	52.3	66.8
RM-R1-7B (Distill)	91.9	**93.7**	28.3	73.2	53.3	85.8	74.8	66.7	**89.4**	**56.3**	64.4	70.7
RM-R1-7B (Instruct)	81.7	84.1	42.6	67.8	56.7	72.7	74.7	**67.0**	89.1	55.9	**64.8**	68.8
FARE-8B	88.1	82.3	**65.8**	68.9	57.0	69.1	88.1	63.2	79.3	55.2	55.4	70.2
RubricRM-8B	93.6	81.7	50.8	77.6	55.4	59.8	86.5	60.9	75.5	52.8	52.4	67.9
DeepSeek-GRM-16B	84.0	69.1	49.4	62.3	51.5	61.7	86.8	55.2	64.3	54.1	53.7	62.9
Our Proposed Reward Models												
SFT-trained												
Base-GRM	91.0	77.2	46.0	81.4	57.1	78.4	86.4	59.9	71.8	52.2	52.5	68.5
Mix-GRM (Breadth)	90.1	72.0	51.1	69.4	54.0	74.1	86.8	60.0	70.3	51.9	52.4	66.6
Mix-GRM (Depth)	89.8	86.1	45.1	75.4	56.2	77.1	81.1	66.8	83.6	54.7	53.7	70.0
Mix-GRM	88.9	87.9	55.7	76.0	55.8	79.6	81.9	63.9	82.1	54.8	54.0	71.0
RLVR-trained												
Base-GRM	93.2	86.4	62.0	77.0	61.7	78.1	**89.5**	64.6	84.1	54.3	50.5	72.9
Mix-GRM (Breadth)	**96.3**	69.3	55.7	71.0	58.0	70.6	86.5	61.7	75.2	53.0	52.8	68.2
Mix-GRM (Depth)	94.6	89.0	61.8	78.7	64.4	81.4	87.4	**67.0**	86.5	55.6	55.5	74.7
Mix-GRM	95.4	89.0	**65.8**	**79.2**	**66.6**	82.5	88.9	65.0	86.7	54.8	55.2	**75.4**

Synthesis Stage	Raw Rationale	Merge B-CoT	Generate D-CoT
Accuracy (%)	87.1	90.2	88.5

Table 12: Solution accuracy tracked across different stages of the synthesis pipeline, demonstrating the stability of the intermediate transformations.

Training Data (9K)	w/o Filtering (Noisy)	w/ Filtering (Verified)
SFT Performance (%)	69.8	70.1

Table 13: SFT performance comparison between noisy and strictly filtered synthesized data.

Training Stage	SFT	RLVR
Structural Alignment (%)	73	95

Table 14: Structural alignment (match rate) of the generated CoT formats with their target domains on the test set across different training stages.

CoT Style	Reasoning Passes	Avg. SFT Tokens	Avg. RLVR Rollouts	Avg. Inference Tokens
D-CoT	2 (Trace + Gen)	624	682	702
B-CoT	2 ($N = 2$ Sample)	711	830	824
Mix-CoT (Ours)	Adaptive	648	725	731

Table 15: Accounting table reporting the compute and token costs. Token consumption across our proposed methods remains within the same order of magnitude, significantly lower than length-scaling baselines like Self-Consistency.

Prompt for Schema Extraction

PRIMARY TASK:
Your mission is to analyze a given reasoning Chain-of-Thought from a generative reward model. From this CoT, you will extract, define, and refine the specific, detailed principles (or rubrics, criteria) it uses to judge the quality of AI-generated responses. For each principle, you must provide a corresponding analysis that traces it directly back to the original text.
INSTRUCTIONS:
You will be given a CoT text below. Please follow these four steps precisely:
1. Deconstruct the CoT: First, perform a close reading of the entire CoT. Identify all explicit evaluation criteria mentioned as well as any implicit judgments or preferences revealed in the model's comparative language.
2. Extract the Core Idea of Each Criterion: For each criterion, do not simply use the high-level category name. Your goal is to uncover the specific description of that criterion as used by the model. Ask yourself: What specific actions, qualities, or content does the model praise or criticize? What makes one response "more accurate" or "clearer" according to this specific CoT?

3. Formulate and Refine the Principle: Convert each core idea you extracted into a formal, normative, and reusable principle.
3.1 Name It: Give the principle a clear and descriptive name that captures its essence (e.g., "Principle of Factual Precision," "Principle of Structural Clarity").
3.2 Define It: Write the principle as a concise, actionable, and universal rule. It should be an instructive statement about what constitutes a high-quality response.
3.3 Be Specific: Avoid vague terms. Instead of "The response should be relevant," specify how it should be relevant based on the CoT's logic, such as "A relevant response must directly and unambiguously address the user's primary question."
3.4 Be Normative: Phrase it as a standard to be met (e.g., "A high-quality response must...").
4. Provide Corresponding Judgment: For each principle you formulate, you must write a brief "CoT Judgment Extraction." To do this, quote or closely paraphrase specific phrases from the CoT that support your formulation.
5. Conclude the sub-verdict in this Judgment: For the principle and corresponding judgment, you should conclude this verdict in this segment.

OUTPUT FORMAT:
You must follow this format strictly for your entire response.
. . .
1. Principle of [Descriptive Name]: [Your refined, normative principle statement.] **Judgment:** [In this principle, what judgment on Response A and Response B quotes or paraphrases from the source CoT.] **Sub-Verdict:** «A/B», In this principle, the judgment judge which assistant Better]
2. Principle of [Descriptive Name]: [Your refined, normative principle statement.] **Judgment:** [In this principle, what judgment on Response A and Response B quotes or paraphrases from the source CoT.]*** **Sub-Verdict:** «A/B», In this principle, the judgment judge which assistant Better]
(Continue this structure for all principles identified in the CoT)
. . .
Extract and Analyse the following CoT Text
{Vanilla-CoT}

Prompt for Breadth-CoT Generation

PRIMARY TASK:
You are provided with a series of lists, each containing Principles, Judgments, and Sub-Verdicts derived from an independent analysis of a Chain-of-Thought (CoT). Your mission is to merge these lists into a single, master list of unique evaluation principles.
INSTRUCTIONS FOR MERGING AND SYNTHESIS:
1. **Deduplication and Semantic Grouping:**
* Compare all Principles with the corresponding Judgments across all provided lists. * Identify and group principles that are **semantically similar**, even if they use different wording (e.g., “Principle of Precision” and “Principle of Correcteness” are likely the same concept).
2. **Principle Refinement:**
* For each semantic group, synthesize the most concise, actionable, and specifically-detailed statement for the **Principle Description**. * Select the most descriptive and formal **Name** for the refined principle.

3. **Judgment Synthesis:**
* For the refined principle, create a new, synthesized **Judgment** block. This block should consist of a curated selection of the most illustrative quotes and paraphrases from the original Judgments across all source lists that led to the consolidated principle. This new Judgment serves as the combined evidence for the principle.
4. **Merge Count:**
* **COUNT THE SOURCES:** For each synthesized principle, you must count the total number of original, distinct principles/judgments from the source lists that were merged to create it. This number is the **Merge Count**.
5. **Sub-Verdict Aggregation:**
* The final **Sub-Verdict** for the synthesized principle must reflect the aggregated trend. Since the judgments are now synthesized, simply use the majority verdict (e.g., if a principle appeared 4 times with [[B]] and 1 time with [[A]], conclude [[B]]). If the verdicts are balanced (e.g., 2 [[A]] and 2 [[B]]), state **[[MIXED]]**.
6. **Strict Output Adherence:**

* Maintain the exact four-part format for every final entry. The output must be one continuous list of unique, synthesized principles.
SOURCE LISTS TO MERGE:
[Insert List 1 Here]
[Insert List 2 Here]
[Insert List 3 Here]
(Continue for all lists)
OUTPUT FORMAT:
You must follow this exact format for your final, merged response.
***
1. Principle of [Refined, Descriptive Name]: [The synthesized, normative principle statement.] **Judgment:** [A synthesis of the most relevant quotes/paraphrases from the source Judgments that supports this consolidated principle.]*** **Merge Count:** [The total number of original source principles/judgments that were merged to form this entry.] **Sub-Verdict:** «A/B/MIXED», The aggregate verdict for this principle across all CoTs.]

2. Principle of [Refined, Descriptive Name]: [The synthesized, normative principle statement.] **Judgment:** [A synthesis of the most relevant quotes/paraphrases from the source Judgments that supports this consolidated principle.]*** **Merge Count:** [The total number of original source principles/judgments that were merged to form this entry.] **Sub-Verdict:** «A/B/MIXED», The aggregate verdict for this principle across all CoTs.]
(Continue this structure for all unique, synthesized principles)
***

Prompt for Depth-CoT Verification

PRIMARY TASK: Your role is to critically assess the quality of two competing responses (Assistant A and Assistant B) against the user's question, leveraging the expert reasoning as the ultimate ground truth.
MANDATORY NON-BIAS RULES: Avoid all position biases (do not favor the first response presented). Do not allow the length or formatting of the responses to influence your evaluation. Do not favor certain names of the assistants. Be as objective, clinical, and data-driven as possible.
PRINCIPLE-BASED EVALUATION
Given some potential principles, {principles}, you should choose the most critical principle (Preferably one principle, with a maximum of three) from them and then evaluate the two Chatbot responses (A and B) based on the choosed principle. This evaluation must directly reference the deep reasoning to instruction and **__must strictly adhere to the following output format for each principle:__**
EXPERT REASONING: {reasoning}

Principle of [Critical Principle Name]:
Judgment: [Give your specific and detailed evaluation in this principle, and if you are referring the this reasoning, you **MUST** quote using '<Answer>']
Sub-Verdict: «A/B/MIXED», «A» if assistant A is better in this principle, «B» if assistant B is better in this principle, «MIXED» if assistant A and B is Tie.
After providing your complete principle-based evaluation, output your final verdict by strictly following this format: [[A]]if assistant A is better, [[B]]if assistant B is better.