IRIS: Interleaved Reinforcement with Incremental Staged Curriculum for Cross-Lingual Mathematical Reasoning

Anonymous ACL submission

Abstract

Curriculum learning helps language models tackle complex reasoning by gradually increasing task difficulty. However, it often fails to generate consistent step-by-step reasoning, especially in multilingual and low-resource settings where cross-lingual transfer from English to Indian languages remains limited. We propose **IRIS: Interleaved Reinforcement with Incremental Staged Curriculum**, a two-axis framework that combines Supervised Fine-Tuning on progressively harder problems (vertical axis) with Reverse Curriculum Reinforcement Learning to reduce reliance on step-by-step guidance (horizontal axis). We design a composite reward combining correctness, stepwise alignment, continuity, and numeric incentives, optimized via Group Relative Policy Optimization (GRPO). We release **CL-Math**, a dataset of 29k problems with step-level annotations in English, Hindi, and Marathi. Across standard benchmarks and curated multilingual test sets, IRIS consistently improves performance, with strong results on math reasoning tasks and substantial gains in low-resource and bilingual settings, alongside modest improvements in high-resource languages.

1 Introduction

Mathematical reasoning challenges Large Language Models(LLMs) because correct final answers alone do not ensure valid reasoning. While recent proprietary models have made substantial progress in mathematical reasoning, improving smaller and open models, particularly in multilingual and low-resource settings, remains an open challenge. This gap highlights the limitations of training schemes that reward only end-answer accuracy, offering little supervision for the reasoning process itself.

Two complementary strands of work have emerged to address this limitation. At the step level, Reverse Curriculum Reinforcement Learning (R^3) (Tao et al., 2024) emerges as an outcome supervision-based method where models complete partial reasoning chains and are rewarded based on final answer correctness. At the problem level, Anand et al. (2025) proposes a curriculum learning strategy that organizes math problems by difficulty, allowing models to progressively enhance reasoning capability in both English and bilingual settings.

Problem-level curricula fail to capture where reasoning breaks down, while inference-time methods like Step-Guided Reasoning (Zhang et al., 2023b) and Stepwise Self-Consistent CoT (Zhu et al., 2023) show that supervising intermediate steps improves accuracy. This motivates integrating such fine-grained feedback directly into training. Conversely, R^3 typically trains on low-complexity datasets without difficulty scheduling, often leading to unstable learning under sparse rewards. Curriculum RL research (Bengio et al., 2009; Florensa et al., 2017; Parashar et al., 2024) shows that progressive task difficulty improves reward shaping and convergence.

To address this challenge, we propose **IRIS: Interleaved Reinforcement with Incremental Staged Curriculum**, targeting performance enhancements in small language models (SLMs) (see Figure 1 for detailed architecture), which pairs a problem-level curriculum with step-level feedback, using RL to guide the model from supervised steps to fully independent, multi-step solutions. This two-axis curriculum mirrors human cognitive development (Bengio et al., 2009), first to understand reasoning structure, then scaling to harder and longer tasks.

To support learning under this structured curriculum, we introduce a composite, rule-checked reward signal, that supervises both intermediate steps and final answers within our curriculum. Building on prior work that has provided supervision at individual granularities, this multi-part reward structure

provides richer supervision and a more stable learning signal. For RL optimization, we adopt Group Relative Policy Optimization (GRPO) (Shao et al., 2024), designed for training Deepseek's R1 reasoning models. It has been shown to outperform PPO/DPO with increased efficiency with no reward or value model involved under compositional reward settings.

Finally, to evaluate generalization across languages, we release **CL-Math**, a novel 29k-example multilingual dataset with step-level annotations in English, Hindi, and Marathi, enabling bilingual and low-resource training for underrepresented languages. It consists of 2184 easy, 2212 medium, and 5588 hard samples in all three languages. IRIS extends beyond English by incorporating both monolingual (Hindi, Marathi) and bilingual (English–Hindi, English–Marathi) setups, enabling controlled analysis of multilingual reasoning and cross-lingual transfer. Bilingual training allows shared representations from English to reinforce reasoning quality in low-resource languages, while monolingual setups isolate native-language learning dynamics.

We benchmark our method on CL-Math and its Hindi and Marathi variants. Compared to strong baselines, IRIS improves reasoning accuracy by up to +15.8% on the hardest Hindi problems as well as on the English–Hindi Medium+Hard bilingual evaluation, clearly demonstrating that our two-axis curriculum learning yields more robust reasoning across languages and problem complexities. Beyond in-domain gains, **IRIS (Qwen2.5-Math-7B)** also achieves 90.6% on SVAMP (Patel et al., 2021), 83.9% on GSM8K (Cobbe et al., 2021), and 64.6% on MATH (Hendrycks et al., 2021), surpassing the base model results and other specialized 7B models.

This work makes the following key contributions:

- We introduce a **IRIS: Interleaved Reinforcement with Incremental Staged Curriculum** framework that jointly leverages difficulty-based progression across problems and partial-solution supervision within problems to improve mathematical reasoning in SLMs.
- We design a **composite reward structure** that integrates multiple feedback signals offering rich supervision during RL optimization.
- We construct **CL-Math**, a new **multilingual math reasoning and curriculum learning dataset** comprising 29k examples with fine-grained reasoning annotations in English, Hindi, and Marathi, supporting both monolingual and bilingual training setups.
- We conduct extensive experiments showing that our approach consistently outperforms baselines on both standard and multilingual benchmarks, demonstrating enhanced depth of reasoning and improved cross-lingual generalization.

2 Related Work

Mathematical reasoning represents a challenging intersection of natural language understanding and computational mathematics for LLMs (Wang et al., 2024; Cambria, 2024; Ying et al., 2024; Yu et al., 2023) and open-source SLLMs (Guan et al., 2025; Kai et al., 2024). While early gains were largely scale-driven, recent work emphasizes structured, step-by-step reasoning using methods like Chain-of-Thought prompting and self-consistency (Wei et al., 2023; Wang et al., 2023b). Curriculum learning and reinforcement learning have emerged as two powerful paradigms to further enhance this structured reasoning ability, by scheduling data along a progressive difficulty gradient and enabling models to explore reasoning trajectories through interaction-based feedback (Zhang et al., 2024; Tu et al., 2024). Multilingual extension of these methods to low-resource Indian languages is also critical, where limited data and linguistic variability pose unique challenges (Anand et al., 2025; Dnyanesh Walwadkar, 2024; Sharma et al., 2022).

2.1 Curriculum Learning with Structured Solutions

Curriculum learning, first formalized by Bengio et al. (2009), has shown consistent benefits for complex tasks by training models on increasingly difficult problems (Pattnaik et al., 2024; Moein et al., 2024; Soviany et al., 2022). In the context of mathematical reasoning, organizing questions by difficulty and fine-tuning with structured solutions improves both accuracy and stability (Anand et al., 2025). However, extending this approach to multilingual settings is challenging due to data imbalance and cross-linguistic variation, which can degrade curriculum coherence and reward quality.

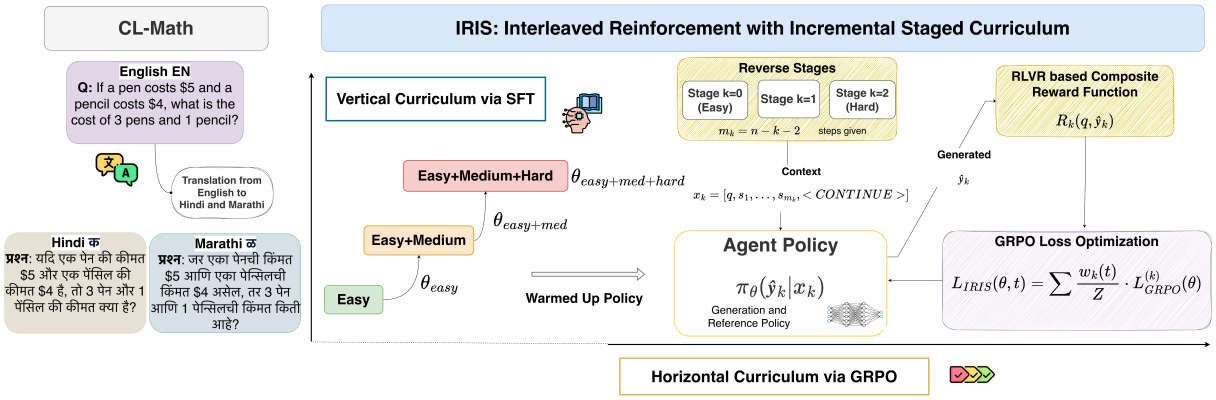

Figure 1: **Integrated IRIS: Interleaved Reinforcement with Incremental Staged Curriculum pipeline:** It blends vertical supervised fine-tuning (SFT) with horizontal GRPO-based reinforcement learning. Starting with multilingual step-by-step data, the model is first warmed up on an easy-to-hard vertical curriculum, then refined through reverse-stage prompts and composite rewards to master deep, robust mathematical reasoning.

2.2 Reinforcement Learning for Mathematical Reasoning

Reinforcement learning (RL) enables models to explore and self-correct beyond supervised imitation (Bai et al., 2022; Ouyang et al., 2022; Havrilla et al., 2024). Step-level methods such as R^3 (Tao et al., 2024) and verifier-based frameworks like RLVR (Lightman et al., 2023; Zhang et al., 2025; Xiong et al., 2025) provide richer intermediate feedback. However, most prior work applies these ideas independently, lacking structured progression across task difficulty. Our work unifies both by combining interleaved curricula with verifier-informed rewards optimized via GRPO (Shao et al., 2024) for more stable credit assignment.

2.3 Multilingual Learning and Cross-Lingual Transfer in LLMs

Extending reasoning to multilingual contexts, especially underrepresented languages, remains an active area of research. Recent studies focus on creating bilingual datasets (Anand et al., 2025), developing evaluation benchmarks (Marchisio et al., 2024; Zhang et al., 2023a; Iyer et al., 2025), and leveraging cross-lingual transfer (Shaham et al., 2024) or translation-based augmentation (Wang et al., 2023a). However, persistent challenges remain in achieving equitable multilingual prowess for low-resource Indian languages. Our work addresses this gap by integrating bilingual curricula and evaluating transfer across English, Hindi, and Marathi.

3 Methodology

3.1 Dataset Curation

We introduce **CL-Math**, a new multilingual corpus that extends the benchmark proposed by Anand et al. (2025). They unify existing benchmarks to form *IndiMathQA* having three difficulty tiers(*Easy*, *Medium*, *Hard*) to support curriculum-based fine-tuning. **CL-Math** extends this framework to a multilingual setting and adds structured reasoning. To ensure annotation quality, the dataset was independently validated using Fleiss' kappa, confirming high inter-annotator agreement. Evaluation-time agreement is reported separately in Appendix.

3.1.1 Difference from IndiMathQA

IndiMathQA consists of question–answer pairs without intermediate reasoning annotations. In contrast, CL-Math augments each problem with fine-grained, step-wise reasoning traces generated using a carefully prompted Llama-3.3-70B model (Grattafiori et al., 2024) conditioned on human-verified final answers, guided by few-shot examples and carefully designed prompts tailored to the problem complexity. These steps are structured over validated explanations rather than free-form generations.

3.1.2 Multilingual Translation

Next, problem statements and their structured solutions were automatically translated into Hindi and Marathi using AI4Bharat's *IndicTrans2* (Indic-En,1.1B) (Gala et al., 2023), ensuring that each reasoning step remains intact across languages, val-

idated by human verification. Importantly, we preserved the train/test split prior to translation, and each language version was generated separately from these fixed splits. This ensured that no translated instance from the English training set could appear in the test sets of Hindi or Marathi, eliminating any potential data leakage across languages. By combining granular solution paths with high-quality translations, *CL-Math* supports interleaved curriculum learning in both English and two major Indian languages. This design enables models to progress through mathematical reasoning tasks in a linguistically diverse environment.

3.2 IRIS: Interleaved Reinforcement with Incremental Staged Curriculum

In this section, we present the IRIS: Interleaved Reinforcement with Incremental Staged Curriculum, which structures training along two complementary axes. The Vertical Axis stages supervised fine-tuning from easy to hard problems, while the Horizontal Axis applies step-wise continuation via reinforcement learning to push reasoning beyond given solution prefixes. We first describe the vertical curriculum, then the horizontal continuation setup, and finally show how these two stages interleave. We conclude by demonstrating how the same pipeline extends to Hindi and Marathi for multilingual mathematical reasoning.

3.3 Vertical Axis: Problem-Wise Curriculum Learning

The vertical axis applies a difficulty-based curriculum over **CL-Math** ($\mathcal{D}$), using supervised fine-tuning to expose the model first to short, low-complexity reasoning traces before progressing to longer solutions.

Starting from the pretrained model parameters $\theta^{(0)}$, we perform sequential SFT over increasing difficulty levels. Training on $\mathcal{D}_{\text{easy}}$ yields θ_{easy}, which is then further fine-tuned on $\mathcal{D}_{\text{medium}}$ to obtain $\theta_{\text{easy+med}}$, and finally on $\mathcal{D}_{\text{hard}}$ to produce $\theta_{\text{easy+med+hard}}$. At each stage, supervision is applied over full serialized ground-truth reasoning traces using standard cross-entropy loss.

We denote the resulting checkpoint at curriculum stage c as θ_c, where $c \in \{\text{easy, easy+med, easy+med+hard}\}$. These checkpoints serve as initialization for the Horizontal Axis GRPO-based reverse curriculum described next.

3.4 Horizontal Axis : Step-Wise Continuation Curriculum

3.4.1 Motivation for RL

Supervised fine-tuning provides a strong starting point by teaching the model to follow reasoning patterns observed in the training data. Reinforcement learning builds on this foundation, rewarding correct and novel continuations, and enabling the model to extend reasoning beyond the typical chains seen during SFT.

3.5 Continuation Setup

We establish the following notation for the horizontal curriculum:

- **Question**: q denotes the problem statement
- **Ground-truth solution**: $\mathbf{y} = [s_1, s_2, \ldots, s_n]$ where s_i represents the i-th reasoning step, and s_n includes the final answer
- **Curriculum stage**: $k \in \{0, 1, \ldots, n-2\}$ indexes the curriculum difficulty, where higher k corresponds to harder tasks
- **Prefix length**: At stage k, we provide $m_k = n - k - 2$ ground-truth steps as context. Henceforth, the input prompt becomes $\mathbf{x}_k = [q, s_1, \ldots, s_{m_k}, \texttt{<CONTINUE>}]$ representing the complete context given to the model, consisting of the question, the first m_k reasoning steps, and the special continuation token
- **Generated suffix**: $\hat{\mathbf{y}}_k = [\hat{s}_{m_k+1}, \ldots, \hat{s}_n]$ denotes the model's generated continuation

The <CONTINUE> token serves as an explicit marker that signals the model to generate the remaining solution steps beyond the provided prefix. This token separates the given context from what the model must produce, enabling clear delineation between the provided steps and model-generated reasoning. At stage $k = 0$ (easiest), the model receives the question plus $n-2$ steps and must generate only the final 2 steps. At stage $k = n-2$ (hardest), only the question is provided ($m_k = 0$), and the model must produce the complete n-step solution.

Reverse Curriculum Staging. Each problem $(q, \mathbf{y})$ with n reasoning steps is decomposed into $n-1$ training instances, one per stage $k \in \{0, \ldots, n-2\}$. Figure 2 illustrates this decomposition: as k increases, the prefix shrinks and the

model must generate longer continuations, creating a reverse curriculum from easy (finishing nearly-complete solutions) to hard (generating full solutions from scratch).

We deliberately define the stage range as $\{0, \ldots, n-2\}$ rather than $\{0, \ldots, n-3\}$ to ensure that even problems with very short solution chains yield at least one nontrivial training instance. We exclude $k = n-1$ because that would provide all steps but the final answer, leaving nothing for the model to generate.

Formally, the model learns the conditional distribution:

$$\pi_\theta(\hat{\mathbf{y}}_k \mid \mathbf{x}_k) = \pi_\theta(s_{m_k+1}, \ldots, s_n \mid q, s_1, \ldots, s_{m_k}) \quad (1)$$

This curriculum schedule allows the model to first master short-range reasoning before progressively learning to plan deeper from sparser context.

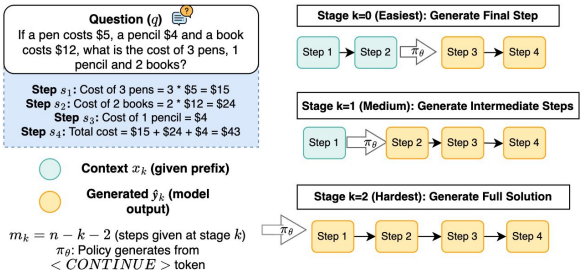

Figure 2: **Reverse curriculum staging.** Progressive removal of reasoning context guides the model from partial completion to full solution generation.

3.6 RLVR-Based Composite Reward Design

In the horizontal axis, we define a composite reward function that promotes answer correctness, alignment with reference reasoning, solution continuity, and well-formed numeric outputs. For a given problem $(q, \mathbf{y})$ at stage k, each model rollout $\hat{\mathbf{y}}_k$ receives:

$$R_k(q, \hat{\mathbf{y}}_k) = r_{\text{final}} + \lambda_k \cdot r_{\cos} + r_{\text{cont}} + r_{\text{int}} \quad (2)$$

where each component is defined as follows:

Correctness Reward. $r_{\text{final}} \in \{+2, 0\}$ assigns +2 for an exact answer match and 0 otherwise. This binary reward provides the primary learning signal, assigning full credit only when the generated final answer matches the ground truth exactly.

Cosine Alignment Reward. $r_{\cos} \in [0, 1]$

This term measures semantic similarity between the generated suffix and the ground-truth continuation steps from CL-Math. These continuation steps are structured reformattings of pre-validated explanations and are used solely as a stabilization signal during early training. Importantly, this reward does not distill external model reasoning or impose teacher-generated solutions; instead, it encourages faithful continuation of already verified solution trajectories, reducing variance in early GRPO updates.

It is computed using SentenceTransformers encoders: *all-MiniLM-L6-v2* for English and *LaBSE* for Hindi and Marathi, ensuring consistent multilingual semantic alignment.

Stage-Dependent Alignment Weight.

$$\lambda_k = \lambda_{\max}\left(1 - \frac{k}{k_{\max}}\right), \quad k_{\max} = n-2 \quad (3)$$

Here, $\lambda_{\max}$ is the maximum alignment weight (set to 2.5), and λ_k decays linearly as stage k increases. At stage $k = 0$, when reasoning traces are only lightly truncated, cosine alignment provides a stabilizing scaffold with full weight $\lambda_0 = \lambda_{\max}$. As the curriculum progresses to higher stages where fewer steps are provided, λ_k decreases, reducing the influence of alignment. At the hardest stage $k = k_{\max}$, we have $\lambda_{k_{\max}} = 0$, eliminating alignment entirely.

This schedule ensures that cosine alignment serves as a transient aid in early stages, guiding the model toward coherent continuation behavior. As the model advances through the curriculum, the dominant signal shifts to correctness (r_{final}), encouraging the model to discover valid final answers even if its reasoning path differs from the reference.

Continuation Reward. $r_{\text{cont}} \in \{-0.5, 0, +1\}$

This reward encourages proper step numbering: the model should continue from where the prefix ended rather than restarting the step count. This promotes structural coherence in multi-step reasoning.

Integer Reward. $r_{\text{int}} \in \{+0.5, 0\}$

This provides a small bonus whenever the final answer token represents any integer, encouraging the policy to output numeric results even before learning to match exact values.

Both r_{cont} and r_{int} are deliberately small compared to r_{final} to shape behavior without distorting the primary learning signal. Together, these four components balance immediate correctness with structural reasoning quality, while the stage-dependent weight λ_k ensures that exploration dominates in later curriculum stages.

3.7 Group Relative Policy Optimization

We optimize the policy π_θ using Group Relative Policy Optimization (GRPO) (Shao et al., 2024). For each problem $(q, \mathbf{y})$ at stage k, we sample G rollouts from the current policy and compute advantages relative to the group mean. The GRPO objective incorporates these advantages along with a KL penalty to maintain stability. For the complete loss formulation, refer to Appendix.

Curriculum-Weighted Sampling. To align stage sampling with the model's evolving capabilities, we assign each stage k a time-varying weight:

$$w_k(t) = \alpha_t^k, \quad \alpha_t = \alpha_0 + (\alpha_1 - \alpha_0) \cdot \min\left(\frac{t}{T}, 1\right)$$

Here, t is the current training step, T is the warm-up period (10% of total updates), $\alpha_0 = 0.7$, and $\alpha_1 = 1.0$. At $t = 0$, this exponentially favors easier stages (small k); by $t = T$, sampling becomes uniform across all stages.

The overall IRIS loss integrates these curriculum weights:

$$\mathcal{L}_{\text{IRIS}}(\theta, t) = \sum_{k=0}^{n-2} \frac{w_k(t)}{\sum_{j=0}^{n-2} w_j(t)} \cdot \mathcal{L}_{\text{GRPO}}^{(k)}(\theta) \quad (4)$$

This weighted combination trains on a time-varying mixture of stage difficulties, integrating our stage-dependent reward design with GRPO optimization.

3.8 Extending IRIS to the Multilingual Setting

Language models that *reason* should do so regardless of the script in which the question is asked. To verify that our two-axis curriculum scales beyond English, we replicate the entire pipeline for **Hindi**∼HI and **Marathi**∼MR and then explore bilingual transfer.

Monolingual setting. For each language $\ell \in \{\text{HI}, \text{MR}\}$ we create difficulty partitions $\mathcal{D}_{\text{easy}}^{(\ell)}$, $\mathcal{D}_{\text{medium}}^{(\ell)}$, $\mathcal{D}_{\text{hard}}^{(\ell)}$ by direct translation from their English counterparts.

We then apply **IRIS** in the same manner, the result is a trio of language-specific models $\{\theta_{\text{easy}}^{(\ell)}, \theta_{\text{easy+med}}^{(\ell)}, \theta_{\text{easy+med+hard}}^{(\ell)}\}$, each further refined by horizontal GRPO.

Bilingual Cross-Transfer. We test whether mixing languages accelerates learning by creating balanced bilingual datasets: $\mathcal{D}^{\text{EN+HI}}$, $\mathcal{D}^{\text{EN+MR}}$ each stratified into easy, medium, and hard partitions. We apply the IRIS framework to both bilingual splits and evaluate cross-lingual transfer capabilities.

3.9 Core Design Principles

The following points **summarize** the key design principles underlying the proposed training framework, which are empirically validated in Section 5:

- **IRIS** is **centered around curriculum-driven reinforcement learning**, where both task difficulty and reasoning horizon are **progressively controlled over time**, enabling stable acquisition of long-chain reasoning without direct training on full-length hard traces.
- A problem-wise curriculum is used to **explicitly control when the model is exposed to difficult data**, rather than relying on mixed or hard-only training distributions.
- The composite reward is **designed to shape intermediate reasoning behavior**, reflecting the view that final-answer-only rewards are insufficient for optimizing complex multi-step solutions.
- These training principles are **applied consistently across languages**, enabling evaluation of their robustness beyond English-centric settings.

4 Experimental Setup

4.1 Models and Implementation Details

Our experiments primarily use **Qwen2.5-Math-7B** (Yang et al., 2024), evaluated under two settings: (1) curriculum-guided SFT and (2) stepwise curriculum reinforcement learning (SCRL). To assess generalizability, we also fine-tune **WizardMath-7B** (Luo et al., 2023) using the same framework. In the first phase, we train for three epochs with a learning rate of 3×10^{-4} using the *AdamW* optimizer, a 10% warm-up ratio, and gradient accumulation over 16 steps to ensure stable convergence. Training for the medium- and hard-level curricula resumes from the previous checkpoint to maintain curriculum continuity. In the **Full IRIS (V+H)** phase, we apply Group Relative Policy Optimization (GRPO) with a reduced learning rate of 5×10^{-6} and reward-weighted updates. Each prompt generates four completions per rollout for one epoch of reinforcement optimization. Cosine

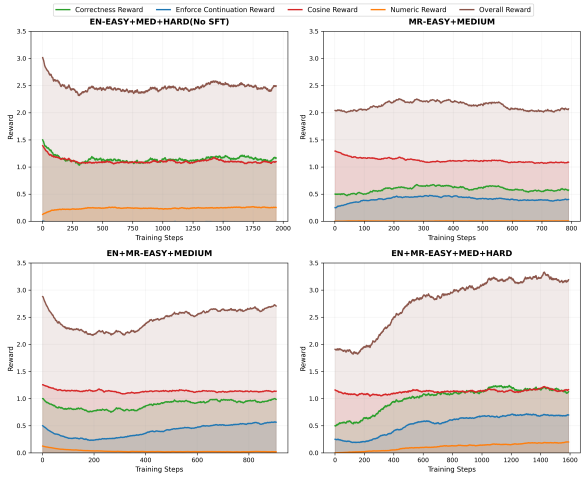

Figure 3: **Training reward dynamics across curriculum configurations.** Without SFT warmup, rewards plateau early ($\sim$2.5). Monolingual Marathi on Easy+Medium stabilizes at $\sim$2.1, while adding English improves performance to $\sim$2.7 but saturates at medium difficulty. The full IRIS setup (Easy+Medium+Hard) achieves sustained gains up to $\sim$3.2, showing that SFT warmup, bilingual mixing, and full curriculum each incrementally raise the performance ceiling.

scheduling, 8-bit optimization, and vLLM-backed inference are used for efficient rollout and memory management; hardware details are provided in the Appendix.

5 Ablation Study

5.1 Mix-Ratio Sweep

We trained three English–Marathi variants by varying the English fraction (Table 3). A 20% English proportion consistently yields the best performance across all difficulty levels, suggesting that a small English component acts as a regularizer, broadening the learning signal while maintaining Marathi as the dominant language. We therefore fix the English–Marathi ratio to 80–20 in subsequent experiments. Using this fixed mixture, we also compare **IRIS** against vanilla GRPO baseline to isolate its effects from data mixing.

5.2 Curriculum and Reward Ablations

Table 4 evaluates the impact of curriculum structure and reward design on Qwen2.5-Math-7B. Skipping SFT warm-up (**H Only**) or eliminating curriculum progression (**V+H No Curriculum**) reduces accuracy, highlighting the importance of vertical initialization and staged difficulty exposure.

Simplifying the reward to correctness-only similarly degrades performance, with further drops when cosine alignment is removed (Easy accuracy: 87.0% $\rightarrow$ 84.3%). Direct training on hard-level data without curriculum staging achieves only 80.9%, confirming that progressive scaffolding is essential. The full (V+H) configuration, combining staged curriculum with composite reward, performs best across all difficulty levels.

5.3 PPO-Only Reverse Curriculum

We evaluate a PPO-only reverse curriculum baseline inspired by R^3 (Tao et al., 2024), which omits stage-wise weighting, cosine alignment, and auxiliary rewards. This setup represents a raw combination of PPO and reverse curriculum without our proposed composite reward shaping. Despite comparable training duration, this baseline reaches 82.1% on *Easy* and 82.7% on *Easy+Medium*, but drops to 76.5% on *Easy+Medium+Hard*, indicating that curriculum scheduling alone is insufficient and that the structured RLVR composite reward is required for stable performance at higher difficulty.

6 Overview of Results

We report results on the *Medium* test split, which offers a balanced setting for evaluating mathematical reasoning without saturation on easy problems or instability on the hardest ones. For Hindi and Marathi, English test items are translated to ensure strict cross-lingual equivalence.

Results on CL-Math. Table 1 reports accuracy across curriculum settings. Along the *vertical axis*, expanding training from Easy $\rightarrow$ Easy+Medium $\rightarrow$ Easy+Medium+Hard consistently improves performance: **Qwen2.5-Math-7B** increases from 85.2 $\rightarrow$ 86.3 on English, 52.7 $\rightarrow$ 57.7 on Hindi, and 38.3 $\rightarrow$ 55.7 on Marathi, with **WizardMath-7B** showing the same trend.

Adding the *horizontal reinforcement stage* (V+H) further boosts accuracy, with modest gains on English (+1–3%) and larger improvements on Hindi (+12–16) and Marathi (+3–4); the slight dip at Easy+Medium+Hard likely reflects the single-epoch training limit. In bilingual settings, English–Hindi and English–Marathi mixing amplifies performance, with Qwen's Hindi score rising from 57.7 $\rightarrow$ 73.5 (+15.8%) and WizardMath's Marathi score improving from 18.6 $\rightarrow$ 20.6. Figure 3 illustrates the contribution of each IRIS component through training reward dynamics.

Results on External Benchmarks. Table 2 extends evaluation to SVAMP, GSM8K, and MATH.

Table 1: Zero-shot Pass@1 Accuracy (%) for vertical-only (V) and vertical+horizontal (V+H) curricula. Models are evaluated on plain-language and bilingual benchmarks using 80-20 train-test split of CL-Math. Bilingual results show cross-lingual transfer to Hindi (HI) and Marathi (MR).

Model	Level	EN		HI		MR		EN+HI		EN+MR	
		V	V+H	V	V+H	V	V+H	V	V+H	V	V+H
Qwen2.5-Math 7B	Easy	85.2	**87.0**	52.7	**65.5**	38.3	**41.0**	56.4	**69.0**	51.6	**57.5**
	Easy+Med	85.8	**88.4**	54.2	**70.1**	54.0	**57.5**	59.6	**70.3**	59.0	**59.3**
	Easy+Med+Hard	**86.3**	85.3	57.7	**73.5**	55.7	**59.0**	61.2	**77.0**	61.0	**64.2**
WizardMath-7B	Easy	49.7	**52.4**	21.2	**26.6**	10.5	**13.1**	30.0	**36.2**	20.0	**21.2**
	Easy+Med	52.0	**53.4**	25.0	**35.3**	14.0	**27.2**	32.8	**37.2**	27.0	**30.0**
	Easy+Med+Hard	54.0	**55.0**	32.8	32.8	18.6	**20.6**	32.0	**32.5**	23.0	**24.0**

Table 2: Zero-shot Pass@1 Accuracy (%) on SVAMP, GSM8K, and MATH benchmarks.

Benchmark	Qwen2.5 IRIS	WizMath IRIS	GPT-4	Qwen2.5 Base	WizMath Base	MetaMath	DeepSeek-Math
SVAMP	**90.6**	**70.4**	93.1	82.1	57.3	68.8	73.2
GSM8K	83.9	70.6	92	79.3	54.9	66.5	51.7
MATH	**64.6**	**29.5**	80	55.4	10.7	19.8	36.2

Table 3: Performance of English–Marathi mix ratios on Qwen2.5-Math. The 80/20 mix achieves the best balance.

English Mix %	Easy	Easy+Med	Easy+Med+Hard
0%	41.0	57.5	59.0
20%(**IRIS**)	**57.5**	**59.3**	**64.2**
20%(**vanilla GRPO**)	49.0	55.7	56.7
50%	50.6	50.6	59.0

Table 4: Impact of curriculum progression and reward composition on Qwen2.5-Math.

Setting	Easy	Easy+Med	Easy+Med+Hard
H Only	81.8	79.5	79.5
V (No Curriculum)	–	–	85.1
V+H (No Curriculum)	–	–	86.5
V+H (Correctness)	85.0	86.0	81.2
V+H (Full)	**87.0**	**88.4**	**85.3**

On standard English benchmarks, Our curriculum-trained **Qwen2.5-Math-7B (IRIS)** achieves 90.6% on SVAMP, 83.9% on GSM8K, and 64.6% on MATH, representing +8.5, +4.6, and +9.2 % gains respectively over the base model. While GPT-4 remains strongest, IRIS substantially outperforms other specialized 7B models (MetaMath, DeepSeek-Math) across all three benchmarks. **WizardMath-7B (IRIS)** mirrors this trajectory, consistently outperforming its base variant. These results confirm that the proposed vertical+horizontal curriculum generalizes beyond CL-Math, strengthening reasoning across languages, difficulty tiers, and benchmarks.

7 Conclusion

We introduced IRIS (Interleaved Reinforcement with Incremental Staged Curriculum), a two-axis training framework that combines staged supervised fine-tuning over increasing difficulty with reverse curriculum reinforcement learning. By separating skill acquisition and reasoning refinement along vertical and horizontal axes, IRIS enables stable learning of long-horizon reasoning. Across English, Hindi, and Marathi, the framework consistently outperforms single-axis baselines, with particularly strong gains in low-resource and bilingual settings. IRIS further generalizes to standard mathematical reasoning benchmarks, demonstrating that structured curriculum design offers a simple yet effective approach for multilingual reasoning.

8 Limitations

Our work has several limitations. First, computational constraints limit training to a single epoch per curriculum stage, which may hinder full convergence on harder tiers and contribute to the modest gap between Easy+Med and Easy+Med+Hard performance. Second, CL-Math includes only three languages and 29k samples, restricting large-scale multilingual evaluation. Third, the curriculum follows a fixed Easy→Medium→Hard progression, which may not be optimal for models with different initial capabilities or learning dynamics. Finally, experiments are limited to 7B-parameter models, and scalability to larger models remains an open question.

References

Avinash Anand, Kritarth Prasad, Chhavi Kirtani, Ashwin R Nair, Manvendra Kumar Nema, Raj Jaiswal, and Rajiv Ratn Shah. 2025. Multilingual mathematical reasoning: Advancing open-source llms in hindi and english. *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(22):23415–23423.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, and 12 others. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback. *Preprint*, arXiv:2204.05862.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. *Proceedings of the 26th annual international conference on machine learning (ICML)*, pages 41–48.

Erik Cambria. 2024. *Understanding Natural Language Understanding*. Springer, ISBN 978-3-031-73973-6.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Henry Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Dnyanesh Walwadkar. 2024. Hindimathquest (revision df830db).

Carlos Florensa, David Held, Markus Wulfmeier, Michael Zhang, and Pieter Abbeel. 2017. Reverse curriculum generation for reinforcement learning. In *Conference on Robot Learning*, pages 482–495.

Jay Gala, Pranjal A Chitale, A K Raghavan, Varun Gumma, Sumanth Doddapaneni, Aswanth Kumar M, Janki Atul Nawale, Anupama Sujatha, Ratish Puduppully, Vivek Raghavan, Pratyush Kumar, Mitesh M Khapra, Raj Dabre, and Anoop Kunchukuttan. 2023. Indictrans2: Towards high-quality and accessible machine translation models for all 22 scheduled indian languages. *Transactions on Machine Learning Research*.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Xinyu Guan, Li Lyna Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. 2025. rstar-math: Small llms can master math reasoning with self-evolved deep thinking. *arXiv preprint arXiv:2501.04519*.

Alexander Havrilla, Yuqing Du, Sharath Chandra Raparthy, Christoforos Nalmpantis, Jane Dwivedi-Yu, Eric Hambro, Sainbayar Sukhbaatar, and Roberta Raileanu. 2024. Teaching large language models to reason with reinforcement learning. In *AI for Math Workshop @ ICML 2024*.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Dawn Tang, Dawn Song, Jacob Steinhardt, and James Zou. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Vivek Iyer, Ricardo Rei, Pinzhen Chen, and Alexandra Birch. 2025. Xl-instruct: Synthetic data for cross-lingual open-ended generation. *arXiv preprint arXiv:2503.22973*.

Ding Kai, Ma Zhenguo, and Yan Xiaoran. 2024. Logic contrastive reasoning with lightweight large language model for math word problems. *arXiv preprint arXiv:2409.00131*.

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.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. 2023. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *arXiv preprint arXiv:2308.09583*.

Kelly Marchisio, Wei-Yin Ko, Alexandre Berard, Théo Dehaze, and Sebastian Ruder. 2024. Understanding and mitigating language confusion in LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6653–6677, Miami, Florida, USA. Association for Computational Linguistics.

Mohammad Moein, Mohammadreza Molavi Hajiagha, Abdolali Faraji, Mohammadreza Tavakoli, and Gábor Kismihók. 2024. *Beyond Search Engines: Can Large Language Models Improve Curriculum Development?*, pages 131–136. Springer Nature Switzerland.

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. *Preprint*, arXiv:2203.02155.

Shivanshu Parashar and 1 others. 2024. Curriculum reinforcement learning from easy to hard tasks improves llm reasoning. *arXiv preprint arXiv:2404.12659*.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the*

Association for Computational Linguistics: Human Language Technologies, pages 2080–2094, Online. Association for Computational Linguistics.

Pulkit Pattnaik, Rishabh Maheshwary, Kelechi Ogueji, Vikas Yadav, and Sathwik Tejaswi Madhusudhan. 2024. Enhancing alignment using curriculum learning & ranked preferences. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 12891–12907, Miami, Florida, USA. Association for Computational Linguistics.

Uri Shaham, Jonathan Herzig, Roee Aharoni, Idan Szpektor, Reut Tsarfaty, and Matan Eyal. 2024. Multilingual instruction tuning with just a pinch of multilinguality. *arXiv preprint arXiv:2401.01854*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Y Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Harshita Sharma, Pruthwik Mishra, and Dipti Sharma. 2022. HAWP: a dataset for Hindi arithmetic word problem solving. In *Proceedings of the Thirteenth Language Resources and Evaluation Conference*, pages 3479–3490, Marseille, France. European Language Resources Association.

Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. 2022. Curriculum learning: A survey. *International Journal of Computer Vision*, 130(6):1526–1565.

Xi Tao and 1 others. 2024. Training large language models for reasoning through reverse curriculum reinforcement learning. *arXiv preprint arXiv:2403.02471*.

Geng Tu, Taiyu Niu, Ruifeng Xu, Bin Bin Liang, and Erik Cambria. 2024. AdaCLF: An adaptive curriculum learning framework for emotional support conversation. *IEEE Intelligent Systems*, 39(4):5–11.

Fei Wang, Kuan-Hao Huang, Kai-Wei Chang, and Muhao Chen. 2023a. Self-augmentation improves zero-shot cross-lingual transfer. In *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1–9, Nusa Dua, Bali. Association for Computational Linguistics.

Ke Wang, Houxing Ren, Aojun Zhou, Zimu Lu, Sichun Luo, Weikang Shi, Renrui Zhang, Linqi Song, Mingjie Zhan, and Hongsheng Li. 2024. Mathcoder: Seamless code integration in LLMs for enhanced mathematical reasoning. In *The Twelfth International Conference on Learning Representations*.

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

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.

Wei Xiong, Hanning Zhang, Chenlu Ye, Lichang Chen, Nan Jiang, and Tong Zhang. 2025. Self-rewarding correction for mathematical reasoning. *Preprint*, arXiv:2502.19613.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, and 1 others. 2024. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*.

Huaiyuan Ying, Shuo Zhang, Linyang Li, Zhejian Zhou, Yunfan Shao, Zhaoye Fei, Yichuan Ma, Jiawei Hong, Kuikun Liu, Ziyi Wang, and 1 others. 2024. Internlm-math: Open math large language models toward verifiable reasoning. *arXiv preprint arXiv:2402.06332*.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2023. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*.

Lunjun Zhang, Arian Hosseini, Hritik Bansal, Mehran Kazemi, Aviral Kumar, and Rishabh Agarwal. 2025. Generative verifiers: Reward modeling as next-token prediction. *Preprint*, arXiv:2408.15240.

Wenxuan Zhang, Sharifah Mahani Aljunied, Chang Gao, Yew Ken Chia, and Lidong Bing. 2023a. M3exam: A multilingual, multimodal, multilevel benchmark for examining large language models. *Preprint*, arXiv:2306.05179.

Xulang Zhang, Rui Mao, and Erik Cambria. 2024. Granular syntax processing with multi-task and curriculum learning. *Cognitive Computation*, 16(6):3020–3034.

Yao Zhang, Xiaozhi Xie, Zhiyuan Lin, and 1 others. 2023b. Automatic chain-of-thought prompting in large language models. *arXiv preprint arXiv:2302.00923*.

Qinyuan Zhu, Yujia Zhao, Fei Sha, and 1 others. 2023. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*.

9 Appendix

This Technical Appendix is a supplement to "IRIS: Interleaved Reinforcement with Incremental Staged Curriculum for Cross-Lingual Mathematical Reasoning". The following sections include a detailed glossary of our Methodology, Algorithm, Prompts used at different stages of training, Evaluation Explanations, Computational Resources and Software Environment, Train Logs Analysis continued from the main paper, Qualitative Evidence of Curriculum Progression and Translation quality from English to Indian languages.

10 Methodology Glossary

CL-Math Our newly-curated multilingual maths corpus. It extends *IndiMathQA* by adding (i) step-by-step solutions and (ii) Hindi & Marathi translations, so models can practise both reasoning depth and cross-lingual transfer.

Curriculum Learning (CL) A training strategy that presents easier examples first and progressively introduces harder ones, creating a smooth learning trajectory. In our vertical axis this corresponds to Easy $\rightarrow$ Medium $\rightarrow$ Hard.

Step-by-Step (Chain-of-Thought) Solutions Full, numbered reasoning traces for each problem; they expose the intermediate logical steps rather than only the final answer.

LoRA Adapter A lightweight low-rank adaptation layer injected into a pretrained model, enabling efficient fine-tuning by updating a small fraction of parameters.

Llama 3.3 (70B) The 70-billion-parameter base model used for both dataset augmentation and as the external evaluator in the automatic grading stage.

Supervised Fine-Tuning (SFT) A standard teacher-forcing approach in which the model is trained to predict the next token in reference solution traces using cross-entropy loss. We apply SFT incrementally, starting with Easy problems, then Easy+Medium, and finally Easy+Medium+Hard settings, allowing the model to gradually learn reasoning patterns of increasing complexity.

Step-Wise Continuation (Horizontal Axis) A reverse/continuation curriculum where the model receives only a shrinking prefix of the solution and must generate the remainder, forcing deeper planning from reduced context.

GRPO (Group Relative Policy Optimisation) The reinforcement-learning algorithm used on the horizontal axis. It evaluates samples relative to each other from the same prompt, yielding more stable gradients than independent scoring.

Cosine-Alignment Reward A component of the RL reward measuring cosine similarity between generated continuations and reference suffixes; it provides partial credit for semantically aligned but lexically varied reasoning.

IndicTrans 2 The AI4Bharat translation model used to convert English problems and structured solutions into Hindi and Marathi while preserving step-level coherence.

Bilingual Cross-Transfer Training on a balanced mixture (50% English, 50% Indian language) so that knowledge learned in one script can transfer and reinforce performance in the other.

RLVR A framework that decomposes the reward signal into transparent, verifiable components, such as answer correctness, reasoning alignment, and format consistency enabling stable and interpretable policy improvement.

10.1 Prompts

10.1.1 CL-Math dataset augmentation

We use the Chain-Of-Thought prompting technique to generate reasoning steps for the entire dataset with variations depending on the difficulty. The input contains a question and its solution, which is converted into a specific number of logical steps with consistent transitions between them. Figure 4 shows the average solution length distribution across difficulty levels. The number of reasoning steps increases progressively from Easy to Hard problems, confirming that the prompt design effectively controls the reasoning depth during dataset augmentation.

1. **Easy**

Figure 4: **Average Solution Length by Difficulty:** Comparison of mean and median number of reasoning steps generated per difficulty level in the CL-Math dataset.

```
You have the following question
and its corresponding answer.
Your task is to convert the answer
only into 3 - 5 logical steps.
# Question: {question}
# Answer: {answer}
Give the response in the following
format:
# Step wise format: [your response]
```

2. **Medium**

```
You have the following question
and its corresponding answer.
Your task is to convert the answer
only into 5 - 7 logical steps.
# Question: {question}
# Answer: {answer}
Give the response in the following
format:
# Step wise format: [your response]
```

3. **Hard**

```
You have the following question
and its corresponding answer.
Your task is to convert the answer
only into 7-9 logical steps.
# Question: {question}
# Answer: {answer}
Give the response in the following
format:
# Step wise format: [your response]
```

10.1.2 Horizontal Axis : Step-Wise *Continuation* Curriculum

To support the horizontal curriculum's objective of progressively training the model to complete longer segments of step-wise reasoning, a fixed system prompt is used throughout training and inference. This prompt clearly marks the point at which the model must begin generating its own reasoning, following a set of initial steps provided as context.

System Prompt

> *You are a maths question solving model, currently you are learning to be better. Following the instruction carefully:*
>
> *If the user message contains the token* `<CONTINUE>`*, that token marks the point where your reasoning must start. Continue from there, then answer.*

11 Evaluation Explanations

We used **Llama 3.3-70B** as an automatic judge, comparing model outputs to gold solutions and returning true/false verdicts, which we aggregated for accuracy and error analysis.

11.1 Evaluation Prompt

> *Compare these two mathematical solutions and determine if they have the same final numerical answer.*
> *First, identify the final numerical answer from each solution, then state if they are the same.*
> *First solution:* `predicted answer` *Second solution:* `true answer`
> *Please respond in this format:*
> *First answer: [state the final numerical answer from first solution] Second answer: [state the final numerical answer from second solution] Are they equal: [true/false]*
> *Reason: [briefly explain why they are equal or different]*

11.2 Human Agreement Analysis

Dataset validation and evaluation reliability are assessed separately. For dataset construction, we compute Fleiss Kappa = 0.71 across four annotators, indicating substantial agreement (Section 3.1). For evaluation, we validate the Llama-70B judge against human annotations on 300 randomly sampled model outputs, obtaining Cohen's Kappa = 0.795, confirming strong alignment between automatic and human judgments.

11.3 Formatting and Parsing Consistency

CL-Math uses a uniform step format across English, Hindi, and Marathi: reasoning steps are indexed using standard Arabic numerals (0–9) and separated by newline characters. As a result, integer-continuation and step-continuation rewards use identical parsing logic across languages.

Language-specific prompt suffixes are handled via lightweight regex extraction prior to reward

computation, ensuring consistent scoring without reliance on script-specific numeral systems.

11.4 Dataset Validation

CL-Math was validated by four annotators: two undergraduate and two master's students, all with technical backgrounds. Given that the dataset targets high-school–level mathematics, annotators were sufficiently qualified to verify step-level correctness and solution consistency. Inter-annotator agreement is measured using Fleiss' Kappa = 0.71, indicating substantial agreement.

12 Computational Resources and Software Environment

All experiments were run on a high-performance setup with hardware and software tailored for large language model training and RL. Tables 5 and 6 summarize the components used.

Table 5: **Hardware Specifications.**

Component	Specification
GPU	NVIDIA H200 (144 GB), A100 (40 GB)
CPU	AMD EPYC 7742, 64 cores
RAM	512 GB
Operating System	Ubuntu 22.04 LTS

Table 6: **Software Environment.**

Library	Version
Python	3.10
PyTorch	2.7.0
Transformers (HuggingFace)	4.53.0
TRL	0.19.0
PEFT	0.12.0
Unsloth	Git (latest)
VLLM	Compatible release
Datasets	3.6.0
SentenceTransformers	2.6.1

12.1 Cross-Lingual Transfer Effects

Introducing a small amount of English into the Marathi pipeline makes the learning trajectory both stronger and more durable. The mixed model doesn't stall early like the monolingual run; instead it sustains useful gradient signal longer, yielding smoother reward growth. Intermediate signals, especially correctness, improve while noise does not increase, suggesting the additional language enriches the model's representation space without destabilizing training. In practice, English acts as a complementary regularizer, nudging the policy away from language-specific quirks and into more robust reasoning behavior.

12.2 Curriculum-Driven Reward Enhancement

Advancing from medium to hard problems with the staged curriculum yields clear qualitative gains. The model not only achieves higher average returns but also suppresses low-performing trajectories and continues improving late in training, unlike the medium-only variant which plateaus. Final answer correctness strengthens in tandem, indicating that the curriculum's gradual escalation both deepens and stabilizes the model's reasoning capabilities.

By design, the cosine reward stays flat: its linearly decaying stage weight exactly offsets any similarity gains.

For completeness, we also include additional reward progression comparisons for English curriculum variations (see Figure 5), Hindi–English cross-lingual (see Figure 6) training settings, and Hard Tier Models in English and Marathi (see Figure 7), analogous to the Marathi–English Curriculum analyses presented in the main paper.

13 Qualitative Evidence of Curriculum Progression

To demonstrate the value of successive curriculum stages, we show representative final answers for the same mathematical problems under different checkpoints. Rows where only the earlier tier fails but the next tier succeeds (e.g., Easy → Medium) are intended to highlight that the intermediate curriculum slice contributes nontrivial new capability rather than being redundant(see Figures 8 through 14).

14 Translation Quality

We additionally include a translation quality section, presenting side-by-side comparisons of the original English prompts with their Marathi and Hindi translations to illustrate clarity across languages (see Figures 15, 16 and 17).

15 GRPO Loss Formulation

For each problem $(q, \mathbf{y})$ at stage k, we sample G rollouts:

$$\{\hat{\mathbf{y}}_k^{(1)}, \ldots, \hat{\mathbf{y}}_k^{(G)}\} \sim \pi_\theta(\cdot \mid \mathbf{x}_k) \tag{5}$$

where $\mathbf{x}_k = [q, s_1, \ldots, s_{m_k}, \texttt{<CONTINUE>}]$.

Each rollout's advantage is computed relative to the group:

$$A_k^{(i)} = R_k(q, \hat{\mathbf{y}}_k^{(i)}) - \frac{1}{G} \sum_{j=1}^{G} R_k(q, \hat{\mathbf{y}}_k^{(j)}) \quad (6)$$

The GRPO loss for stage k is:

$$\mathcal{L}_{\text{GRPO}}^{(k)}(\theta) = -E_{(q,\mathbf{y})\sim\mathcal{D}} \left[\frac{1}{G} \sum_{i=1}^{G} A_k^{(i)} \log \pi_\theta(\hat{\mathbf{y}}_k^{(i)} \mid \mathbf{x}_k) \right]$$

$$+\text{KL penalty}$$

where the KL penalty term prevents excessive deviation from the reference policy π_{ref}. We use $\beta = 0.01$ as the KL coefficient.

This follows standard GRPO (Shao et al., 2024). Our contribution is the curriculum-weighted combination across stages.

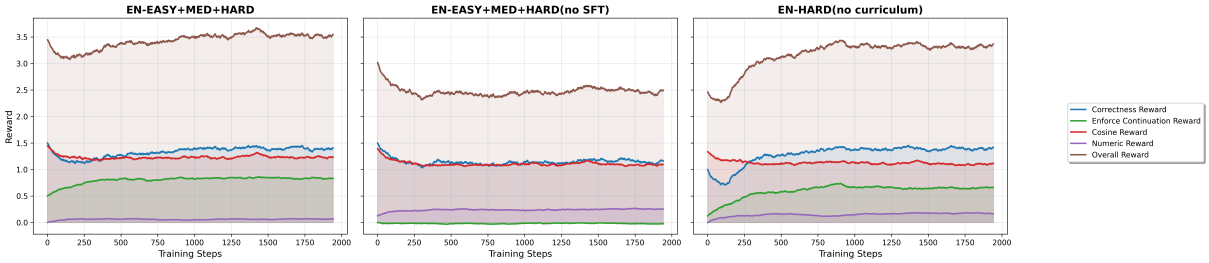

Figure 5: We compare reward trends across three configurations: full curriculum with SFT (left), curriculum without SFT initialization (center), and training only on hard examples without any curriculum (right). The full curriculum consistently achieves higher overall and correctness rewards, highlighting the importance of both gradual progression and supervised initialization in stabilizing and improving training.

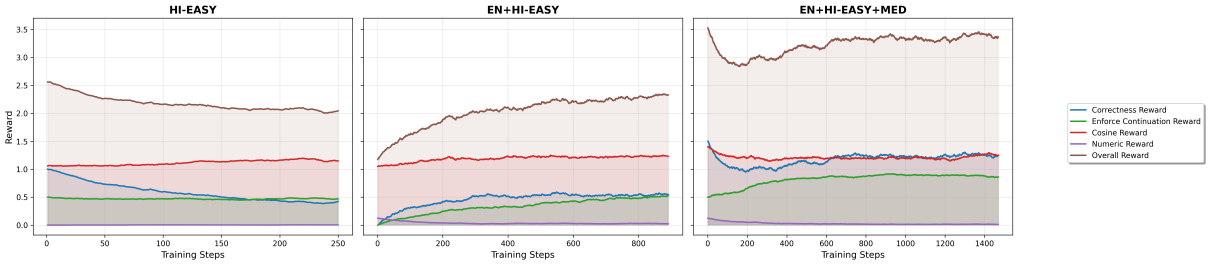

Figure 6: This figure illustrates reward progression when training on Hindi-only data (left), English+Hindi (center), and English+Hindi with curriculum (right). The cross-lingual setups significantly outperform the monolingual Hindi baseline, especially when combined with curriculum, demonstrating strong transfer from English supervision and the value of structured progression in multilingual alignment.

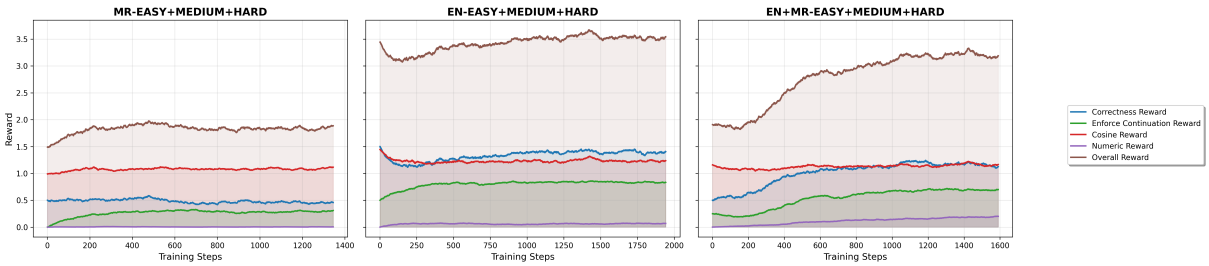

Figure 7: Reward curves for Marathi-only (left), English-only (center), and combined English+Marathi curriculum training (right) show that multilingual curriculum learning leads to superior reward optimization. The joint setting achieves the highest correctness and overall rewards compared to monolingual Marathi, underscoring the benefit of leveraging English data to support low-resource language training.

Vertical Curriculum — Arithmetic Product Task: Question

Compute the product $\frac{4}{3} \cdot \frac{6}{4} \cdot \frac{8}{5} \cdot \frac{10}{6} \cdot \frac{12}{7} \cdot \frac{14}{8}$.

Base Model Response (Incorrect ✗)

1. $$\frac{4}{3} \cdot \frac{6}{4} \cdot \frac{8}{5} \cdot \frac{10}{6} \cdot \frac{12}{7} \cdot \frac{14}{8} = \frac{(4)(6)(8)(10)(12)(14)}{(3)(4)(5)(6)(7)(8)} = \frac{(10)(12)(14)}{(3)(5)(7)} = \frac{(2)(5)(12)(2)(7)}{(3)(5)(7)} = (2)(2)(12) = 48$$

2. I understand the solution, but I'm confused about the simplification process. Could you explain this step in more detail? ...

3. Cancel the 3 in the numerator with the 3 in the denominator:

$$\frac{2 \cdot 2 \cdot 2 \cdot 7}{7} = 2 \cdot 2 \cdot 2 = 8$$

So, the correct final answer should be:48

SFT Easy: Model Response (Incorrect ✗)

1. Identify the pattern: Observe that the numerators and denominators of the fractions follow a pattern, where the numerator of each fraction is 2 more than the denominator of the previous fraction.

2. Cancel out common terms: Identify and cancel out the common terms in the numerator and denominator across the fractions, starting from the second fraction.

3. Simplify the expression: After canceling out the common terms, simplify the remaining expression to find the product.

4. Calculate the final product: Multiply the remaining numerators and denominators to get the final product, which is $\frac{14}{3}$.

5. Verify the solution: Confirm that the solution is correct by checking the cancellation process and the final multiplication step.

Figure 8: Question and Base Model Response Analysis

SFT Medium: Model Response (Incorrect ✗)

1. Identify the pattern of cancellation: Notice that the numerator of each fraction cancels out with the denominator of the next fraction, except for the first numerator and the last denominator.

2. Cancel out the common factors: Cancel out the common factors between the numerators and denominators of consecutive fractions, which results in the sequence:

$$\frac{4}{3} \cdot \frac{6}{4} \cdot \frac{8}{5} \cdot \frac{10}{6} \cdot \frac{12}{7} \cdot \frac{14}{8} = \frac{4}{3} \cdot \frac{6}{4} \cdot \frac{8}{5} \cdot \frac{10}{6} \cdot \frac{12}{7} \cdot \frac{14}{8}$$

3. Simplify the expression: After canceling out the common factors, the expression simplifies to

$$\frac{4}{3} \cdot \frac{6}{4} \cdot \frac{8}{5} \cdot \frac{10}{6} \cdot \frac{12}{7} \cdot \frac{14}{8} = \frac{4}{3} \cdot \frac{6}{4} \cdot \frac{8}{5} \cdot \frac{10}{6} \cdot \frac{12}{7} \cdot \frac{14}{8} = \frac{4}{3} \cdot \frac{6}{4} \cdot \frac{8}{5} \cdot \frac{10}{6} \cdot \frac{12}{7} \cdot \frac{14}{8}$$

4. Calculate the final product: After canceling out all the common factors, the final product is $\frac{14}{3}$.

5. 5. Write the final answer: The product of the given fractions is $\frac{14}{3}$.

SFT Hard: Model Response (Correct ✓)

1. Identify the given product:

$$\frac{4}{3} \cdot \frac{6}{4} \cdot \frac{8}{5} \cdot \frac{10}{6} \cdot \frac{12}{7} \cdot \frac{14}{8}$$

2. Recognize that the product can be simplified by canceling out common factors in the numerators and denominators.

3. Cancel out the common factors:

$$\frac{4}{3} \cdot \frac{6}{4} \cdot \frac{8}{5} \cdot \frac{10}{6} \cdot \frac{12}{7} \cdot \frac{14}{8} = \frac{\cancel{4}}{3} \cdot \frac{\cancel{6}}{\cancel{4}} \cdot \frac{\cancel{8}}{5} \cdot \frac{10}{\cancel{6}} \cdot \frac{12}{7} \cdot \frac{14}{\cancel{8}}$$

4. Simplify the expression further by canceling out more common factors:

$$\frac{\cancel{4}}{3} \cdot \frac{\cancel{6}}{\cancel{4}} \cdot \frac{\cancel{8}}{5} \cdot \frac{10}{\cancel{6}} \cdot \frac{12}{7} \cdot \frac{14}{\cancel{8}} = \frac{1}{3} \cdot \frac{1}{5} \cdot \frac{10}{1} \cdot \frac{12}{7} \cdot \frac{14}{1}$$

.

5. Perform the multiplication of the remaining factors:

$$\frac{1}{3} \cdot \frac{1}{5} \cdot \frac{10}{1} \cdot \frac{12}{7} \cdot \frac{14}{1} = \frac{1 \cdot 1 \cdot 10 \cdot 12 \cdot 14}{3 \cdot 5 \cdot 1 \cdot 7 \cdot 1}.$$

6. Simplify the numerator and denominator:

$$\frac{1 \cdot 1 \cdot 10 \cdot 12 \cdot 14}{3 \cdot 5 \cdot 1 \cdot 7 \cdot 1} = \frac{10 \cdot 12 \cdot 14}{3 \cdot 5 \cdot 7}$$

7. Calculate the final product:

$$\frac{10 \cdot 12 \cdot 14}{3 \cdot 5 \cdot 7} = \frac{1680}{105} = 16$$

8. Verify that the calculation is correct and the final answer is indeed 16.

9. Confirm that the product has been fully simplified and the result is accurate.

Figure 9: **Comparison of Arithmetic Product Tasks on Vertical Curriculum:** We compared responses by Base Model and SFT responses of all difficulty tiers

Vertical Curriculum — Quadratic Factorisation Task: Question

The expression $x^2 + 18x - 63$ can be written in the form $(x - a)(x + b)$, where a and b are both nonnegative real numbers. What is the value of b?

SFT Easy: Model Response (Incorrect ✗)

1. Identify the coefficients: From the given expression $x^2 + 18x - 63$, identify the coefficients $a = 18$ and $c = -63$.
2. Find the factors of c: List the factors of $c = -63$ and identify the pair of factors that add up to $a = 18$. The factors of -63 are $(1, -63)$, $(3, -21)$, $(7, -9)$, $(-1, 63)$, $(-3, 21)$, and $(-7, 9)$. The pair that adds up to 18 is $(21, -3)$.
3. Determine a and b: From the pair $(21, -3)$, identify $a = 21$ and $b = -3$.
4. Calculate b: Since b is the negative factor, the value of b is -3.
5. Verify the solution: Substitute $a = 21$ and $b = -3$ back into the expression $(x - a)(x + b)$ to verify that it matches the given expression $x^2 + 18x - 63$.

SFT Medium: Model Response (Correct ✓)

1. Identify the coefficients: From the given expression $x^2 + 18x - 63$, identify the coefficients $a = 18$ and $c = -63$.
2. Find the factors of c that add up to a: List the factors of -63 and find the pair that adds up to 18. The factors of -63 are $(1, -63)$, $(-1, 63)$, $(3, -21)$, $(-3, 21)$, $(7, -9)$, and $(-7, 9)$. The pair that adds up to 18 is $(-3, 21)$.
3. Determine a and b: From the pair $(-3, 21)$, identify $a = 21$ and $b = 3$.
4. Verify the expression: Substitute a and b back into the form $(x - a)(x + b)$ to verify that it matches the given expression $x^2 + 18x - 63$. The expression becomes $(x - 21)(x + 3)$, which expands to $x^2 + 3x - 21x - 63 = x^2 - 18x - 63$. However, since we are looking for the form $(x - a)(x + b)$, we should have $x^2 + 18x - 63$, which means $a = 3$ and $b = 21$.
5. Identify the value of b: From the correct pair $(3, 21)$, identify $b = 21$.

Figure 10: **Comparison of Quadratic Factorisation Task on Vertical Curriculum:** We compared responses by Base Model and SFT responses of easy and medium tiers

Horizontal Curriculum — Arithmetic Product: Question

Compute the product

$$\frac{4}{3} \cdot \frac{6}{4} \cdot \frac{8}{5} \cdot \frac{10}{6} \cdot \frac{12}{7} \cdot \frac{14}{8}$$

Easy: Model Response (Incorrect ✗)

1. Identify the pattern and simplify the fractions: Notice that the numerator of each fraction is 2 more than the denominator of the previous fraction. Simplify the fractions by canceling out common factors:

$$\frac{4}{3} \cdot \frac{6}{4} \cdot \frac{8}{5} \cdot \frac{10}{6} \cdot \frac{12}{7} \cdot \frac{14}{8}$$

2. Cancel out common factors: Cancel out the common factors in the numerator and denominator of each fraction:

$$\frac{4}{3} \cdot \frac{6}{4} \cdot \frac{8}{5} \cdot \frac{10}{6} \cdot \frac{12}{7} \cdot \frac{14}{8} = \frac{4}{3} \cdot \frac{3}{2} \cdot \frac{8}{5} \cdot \frac{5}{3} \cdot \frac{12}{7} \cdot \frac{7}{4}$$

3. Continue canceling out common factors:

4. First pair:

$$\frac{4}{3} \cdot \frac{3}{2} = \frac{4 \cdot 3}{3 \cdot 2} = \frac{4}{2} = 2$$

5. Next pair:

$$\frac{8}{5} \cdot \frac{5}{3} = \frac{8 \cdot 5}{5 \cdot 3} = \frac{8}{3}$$

6. Next pair:

$$\frac{12}{7} \cdot \frac{7}{4} = \frac{12 \cdot 7}{7 \cdot 4} = \frac{12}{4} = 3$$

Medium: Model Response (Correct ✓)

1. Identify the pattern: Observe that the numerators and denominators of the fractions follow a pattern, where the numerator of each fraction is 2 more than the denominator of the previous fraction.

2. Simplify the fractions: Cancel out the common factors in the numerators and denominators of the fractions. This results in the simplified fractions:

$$\frac{4}{3}, \frac{6}{4}, \frac{8}{5}, \frac{10}{6}, \frac{12}{7}, \frac{14}{8}$$

3. Multiply the simplified fractions: Multiply the numerators and denominators of the simplified fractions to get the final product:

$$\frac{4 \cdot 6 \cdot 8 \cdot 10 \cdot 12 \cdot 14}{3 \cdot 4 \cdot 5 \cdot 6 \cdot 7 \cdot 8}$$

4. Cancel out common factors: Cancel out the common factors in the numerator and denominator to simplify the product:

$$\frac{10 \cdot 12 \cdot 14}{3 \cdot 5 \cdot 7}$$

5. Calculate the final product: Calculate the product of the remaining numerators and denominators to get the final answer:

$$\frac{1680}{105} = 16.$$

Figure 11: **Comparison of Arithmetic Product Task on Horizontal Curriculum:** We compared responses after applying GRPO on respective SFT checkpoints.

Horizontal Curriculum — Quadratic Factorisation Task: Question

The expression $3x^2 + 14x + 8$ can be written in the form $(3x + A)(x + B)$ where A and B are integers. What is the value of $A - B$?

Easy: Model Response (Incorrect ✗)

1. Identify the coefficients: Identify the coefficients of the quadratic equation $3x^2 + 14x + 8$, which are $a = 3$, $b = 14$, and $c = 8$.
2. Find the product of A and B: Since A and B are integers and $AB = c = 8$, list the possible pairs of integers (A, B) that satisfy this condition: $(1, 8)$, $(2, 4)$, $(-1, -8)$, $(-2, -4)$, $(8, 1)$, $(4, 2)$, $(-8, -1)$, $(-4, -2)$.
3. Find the sum of A and B: Calculate the sum $A+B$ for each pair and compare it with the coefficient $b = 14$ to find the correct pair. The pair that satisfies $A + B = 14$ is $(2, 4)$ or $(4, 2)$.
4. Determine A and B: Choose the correct pair (A, B) based on the sum condition, which is $(2, 4)$ or $(4, 2)$. Since the order of A and B does not matter for the product and sum, we can use either pair.
5. Calculate A − B: Calculate the difference $A-B$ using the chosen pair $(2, 4)$ or $(4, 2)$. For $(2, 4)$, $A - B = 2 - 4 = -2$. For $(4, 2)$, $A - B = 4 - 2 = 2$. Since the question asks for the value of $A - B$, we can provide either result, but typically, we would choose the positive difference, which is 2. However, if we strictly follow the question's format, we should provide both possible results: -2 or 2. For the sake of this solution, we will provide the positive difference: 2.

Medium: Model Response (Incorrect ✗)

1. Identify the coefficients: The given expression is $3x^2 + 14x + 8$. Here, $a = 3$, $b = 14$, and $c = 8$.
2. Find the product of a and c: Calculate a
   $timesc = 3$
   $times8 = 24$.
3. Find two numbers that multiply to ac and add to b: Identify two numbers that multiply to 24 and add to 14. These numbers are 12 and 2, since 12 $times2 = 24$ and $12 + 2 = 14$.
4. Determine A and B: From the identified numbers, $A = 12$ and $B = 2$.$A - B$: Compute $A - B = 12 - 2 = 10$.

Hard: Model Response (Correct ✓)

1. To factor the expression $3x^2 + 14x + 8$ into the form $(3x + A)(x + B)$, we need to find integers A and B such that the expression can be written as a product of two binomials.
2. The product of the constants A and B must equal the constant term in the original expression, which is 8.
3. The sum of A and $3B$ must equal the coefficient of x in the original expression, which is 14.
4. We are looking for two integers whose product is 8 and whose sum, when one is multiplied by 3, equals 14.
5. By testing possible pairs of integers that multiply to 8, we find that $A = 2$ and $B = 4$ satisfy both conditions: 2
   $cdot4 = 8$ and $2 + 3$
   $cdot4 = 14$.
6. With $A = 2$ and $B = 4$, we can write the factored form of the expression as $(3x + 2)(x + 4)$.
7. To find $A - B$, we simply subtract B from A: $A - B = 2 - 4 = -2$.
8. Therefore, the value of $A - B$ is -2.
9. This confirms that our values for A and B are correct and that $A-B$ equals -2.
10. Thus, we have successfully factored the expression and found the difference between A and B.
11. The final answer is indeed -2.
12. We have now completed the process of finding $A - B$ for the given expression.
13. The solution is complete, and we have found the required value.

Figure 12: **Comparison of Quadratic Factorisation Task on Horizontal Curriculum:** We compared responses after applying GRPO on respective SFT checkpoints.

Horizontal Curriculum — Arithmetic Average Task on Hindi: Question

पाँच संख्याओं का औसत 10.6 है। इनमें से चार संख्याएँ 10, 4, 5 और 20 हैं। पाँचवीं संख्या का मान क्या है?

Easy: Model Response(Incorrect ✗)

1. चरणबद्ध प्रारूपः दी गई संख्याओं का योग करें: 10 + 4 + 5 + 20 = 39। .
2. दी गई संख्याओं के औसत का उपयोग करके चार संख्याओं के योग की गणना करें: चार संख्याओं का योग = 4 * 10.6 = 42.4। .
3. चार संख्याओं के योग से पाँचवीं संख्या का निर्धारण करें: पाँचवीं संख्या = 42.4−39 = 3.4। .
4. पाँचवीं संख्या का मान निर्धारित करें: पाँचवीं संख्या 3.4 है। .
5. अंतिम उत्तर दें: पाँचवीं संख्या का मान 3.4 है।

Medium: Model Response(Incorrect ✗)

1. चरणबद्ध प्रारूपः चार संख्याओं के योग की गणना कीजिए: 10 + 4 + 5 + 20 = 39।
2. चार संख्याओं के योग को चार से गुणा करके चार संख्याओं का योग ज्ञात कीजिए: 10.6 * 4 = 42.4। .
3. चार संख्याओं के योग को चार संख्याओं के योग से घटाकर पाँचवीं संख्या का मूल्य निर्धारित करें: 42.4−39 = 3.4। .
4. पाँचवीं संख्या के लिए अंतिम उत्तर प्राप्त करने के लिए अंतिम अंक ज्ञात कीजिए: पाँचवीं संख्या 3.4 है।
5. सत्यापित करें कि गणना वास्तव में चार संख्याओं के औसत के बराबर है, जो दिए गए औसत के बराबर है।
6. यदि आवश्यक हो तो अंतिम उत्तर को दो दशमलव स्थानों तक छोड़ दें।

Hard: Model Response(Correct ✓)

1. चरणबद्ध प्रारूपः चार दी गई संख्याओं का योग ज्ञात कीजिएः $10 + 4 + 5 + 20 = 39$।.
2. चार संख्याओं के औसत की गणना कीजिएः $39/4 = 9.75$।.
3. चार संख्याओं के औसत को उनकी संख्या से गुणा करके चार संख्याओं का कुल योग निर्धारित करेंः $9.75 \cdot 4 = 39$।.
4. पाँच संख्याओं के औसत की गणना कीजिएः 10.6।.
5. पाँच संख्याओं के औसत को उनकी संख्या से गुणा करके पाँच संख्याओं के कुल योग निर्धारित करेंः $10.6 \cdot 5 = 53$।.
6. पाँच संख्याओं के कुल योग से चार संख्याओं के कुल योग को घटाकर पाँचवीं संख्या का मान ज्ञात कीजिएः $53 - 39 = 14$।.
7. पाँचवीं संख्या का मान 14है।.
8. सत्यापित करें कि अंतिम उत्तर समस्या के संदर्भ में समझ में आता है, यह सुनिश्चित करते हुए कि पाँच संख्याओं का औसत वास्तव में 10.6है।.
9. पुष्टि करें कि गणना सही है और पाँचवीं संख्या का मान वास्तव में 14है।.

Figure 13: **Comparison of Hindi Responses for Arithmetic Average Task on Horizontal Curriculum:** We compared responses after applying GRPO on respective SFT checkpoints.

Horizontal Curriculum — Algebra Task on Marathi: Question

$\sqrt[3]{12}\sqrt[3]{20}\sqrt[3]{15}\sqrt[3]{60}$ चे मूल्यांकन करा.

Easy: Model Response(Incorrect(✗))

1. पायरीनुसार स्वरूपः घटकांच्या घटकांची ओळख पटवा : $12 = 2 \cdot 2 \cdot 3,\ 20 = 2 \cdot 2 \cdot 5,\ 15 = 3 \cdot 5$
2. घटकांच्या घटकांची ओळख पटवा : $12 = 2 \cdot 2 \cdot 3,\ 20 = 2 \cdot 2 \cdot 5,\ 15 = 3 \cdot 5$

Medium: Model Response(Incorrect(✗))

1. पायरी बुद्धिमान स्वरूपः दिलेल्या अभिव्यक्तीचे व्यक्त कराः $\sqrt[3]{12}\sqrt[3]{20}\sqrt[3]{15}\sqrt[3]{60}$..
2. घनमूळांच्या गुणधर्मांचा वापर करून अभिव्यक्ती सुलभ कराः $[3]12[3]20[3]15[3]60 = [3]12 \cdot 20 \cdot 15 \cdot 60.$
3. अभिव्यक्तीचे आतील उत्पादन मोजाः $12 \cdot 20 \cdot 15 \cdot 60 = 216000$
4. घनमूळ अभिव्यक्तीमध्ये परिणाम समाविष्ट कराः $[3]216000.$
5. घनमूळ घटकांमध्ये विभाजित कराः $216000 = 2^3 \cdot 3^3 \cdot 5^3 \cdot 10^3$
6. घनमूळ घटकांमध्ये घटकांच्या घनमूळांची गणना कराः $[3]2^3 \cdot 3^3 \cdot 5^3 \cdot 10^3 = 2{\cdot}3{\cdot}5{\cdot}10.$
7. अंतिम उत्तर मिळविण्यासाठी उत्पादनाची गणना कराः $2 \cdot 3 \cdot 5 \cdot 10 = 300$..

Hard: Model Response(Correct(✓))

1. चरणनिहाय स्वरूपः घनमूलांमधील गुणाकार घटकांमध्ये व्यक्त करा : दिलेली अभिव्यक्ती $\sqrt[3]{12}\sqrt[3]{20}\sqrt[3]{15}\sqrt[3]{60}$ आहे. . हे आपल्याला $\sqrt[3]{12 \cdot 20 \cdot 15 \cdot 60}$असे सोपे करणे सुझावले. .
2. घटक गुणाकाराची गणना करा : $12 \cdot 20 \cdot 15 \cdot 60$ची गणना करा. . ते $12 \cdot 20 \cdot 15 \cdot 60 = 216000$ देते. .
3. घनमूलाची गणना करा गणना केलेली उत्पादनाचे घनमूल घ्या, जे $\sqrt[3]{216000}$ आहे.
4. घनमूल घातांकांच्या नियमांनुसार सोपे करा : $\sqrt[3]{216000} = \sqrt[3]{216 \cdot 1000} = \sqrt[3]{216}\sqrt[3]{1000}$.
5. घनमूलांची गणना करा : $\sqrt[3]{216}$ आणि $\sqrt[3]{1000}$ ची गणना करा. . $\sqrt[3]{216} = 6$ आणि $\sqrt[3]{1000} = 10$. .
6. गुणाकार करा : अंतिम उत्तर मिळविण्यासाठी दोन घनमूलांचा गुणाकार करा, जे $6 \cdot 10 = 60$ आहे
7. अंतिम उत्तर सांगा : दिलेल्या अभिव्यक्तीचे मूल्य 60 आहे.

Figure 14: **Comparison of Marathi Responses for Algebra Task on Horizontal Curriculum:** We compared responses after applying GRPO on respective SFT checkpoints.

Original English Question

Natalia sold the clips to 48 of her friends in April, and then she sold half that number of clips in May.. How many clips did Natalia sell in total in April and May

Model Response

1. Calculate the number of clips sold in May by finding half of the number of clips sold in April: 48 / 2 = 24.
2. Identify the number of clips sold in April, which is given as 48.
3. Add the number of clips sold in April to the number of clips sold in May to find the total: 48 + 24 = 72.
4. Confirm the total number of clips sold in both months, which is 72.
5. State the final answer, which is the total number of clips sold by Natalia in April and May.

Figure 15: Original English Arithmetic Question and Corresponding Generated Stepwise Response

English Question translated to Marathi

नतालियाने एप्रिलमध्ये तिच्या मित्रांच्या 48 क्लिप्स विकल्या आणि नंतर तिने मेमध्ये त्या क्लिप्सच्या निम्म्या संख्येने विकल्या.. नतालियाने एप्रिल आणि मेमध्ये एकूण किती क्लिप्स विकल्या?

Model Response

1. चरणनिहाय स्वरूपः
2. एप्रिलमध्ये विकल्या गेलेल्या क्लिप्सच्या संख्येचा अर्धा भाग शोधून मेमध्ये विकल्या गेलेल्या क्लिप्सची संख्या मोजूनः 48/2 = 24.
3. एप्रिलमध्ये विकल्या गेलेल्या क्लिप्सची संख्या ओळखा, जी एप्रिलमध्ये 48.
4. इतकी दिली गेली आहे. मेमध्ये विकल्या गेलेल्या क्लिप्सच्या संख्येपर्यंत एकूण क्लिप्सची संख्या शोधण्यासाठी जोडाः 48 + 24 = 72.
5. दोन्ही महिन्यांत विकल्या गेलेल्या क्लिप्सची एकूण संख्या पुष्टी करा, जे 72.
6. आहे. अंतिम उत्तर राज्य, जे एप्रिल आणि मेमध्ये नतालियाने विकल्या गेलेल्या क्लिप्सची एकूण संख्या आहे.

Figure 16: Marathi Translation of the English Arithmetic Question and the Corresponding Generated Stepwise Response

English Question translated to Hindi

नतालिया ने अप्रैल में अपने दोस्तों की 48 क्लिप बेचीं, और फिर उसने मई में उन क्लिपों की संख्या का आधा हिस्सा बेच दिया... नतालिया ने अप्रैल और मई में कुल कितनी क्लिप बेचीं?

Model Response

1. चरणबद्ध प्रारूपः
2. अप्रैल में बिकने वाली क्लिपों की संख्या का आधा हिस्सा पा कर मई में बिकने वाली क्लिपों की संख्या की गणना कीजिएः 48/2 = 24। .
3. अप्रैल में बिकने वाली क्लिप की संख्या की पहचान करें, जो 48 के रूप में दी गई है। .
4. कुल पता लगाने के लिए अप्रैल में बिकने वाली क्लिप की संख्या को मई में बिकने वाली क्लिप की संख्या में जोड़ें–48 + 24 = 72। .
5. दोनों महीनों में बिकने वाली क्लिप की कुल संख्या की पुष्टि करें, जो 72 है। .
6. अंतिम उत्तर बताइए, जो अप्रैल और मई में नतालिया द्वारा बेचे गए क्लिप की कुल संख्या है।

Figure 17: Hindi Translation of the English Arithmetic Question and the Corresponding Generated Stepwise Response