

GANITLLM: Difficulty-Aware Bengali Mathematical Reasoning through CURRICULUM-GRPO

Shubhashis Roy Dipta¹, Khairul Mahbub², Nadia Najjar²

¹University of Maryland, Baltimore County

²University of North Carolina at Charlotte

sroydip1@umbc.edu kmahbub@charlotte.edu

Abstract

We present a Bengali mathematical reasoning model called GANITLLM (named after the Bangla word for mathematics, *Ganit*), together with a new difficulty-aware Bengali math corpus and a curriculum-based GRPO pipeline. Bengali is one of the world’s most widely spoken languages, yet existing LLMs either reason in English and then translate, or simply fail on multi-step Bengali math, in part because reinforcement learning recipes are tuned for high-resource languages and collapse under reward sparsity in low-resource settings. To address this, we construct **GANIT**, a rigorously filtered and decontaminated Bengali math dataset with automatic difficulty tags derived from the pass@k of a strong evaluator model. Building on this dataset, we propose **CURRICULUM-GRPO**, which combines multi-stage training (SFT + GRPO) with difficulty-aware sampling and verifiable rewards for format, numerical correctness, and Bengali reasoning. On Bn-MGSM and Bn-MSVAMP, GANITLLM-4B improves over its Qwen3-4B base by **+8** and **+6** accuracy points, respectively, while increasing the percentage of Bengali reasoning tokens from **14%** to over **88%** and reducing average solution length from **943** to **193** words.¹

1 Introduction

Recent Large Language Models (LLMs) show strong reasoning performance across high-resource languages like English (Shi et al., 2023). In contrast, progress in low-resource languages remains limited (Lai and Nissim, 2024). Bengali, the seventh most spoken language worldwide², clearly reflects this gap (Bhowmik et al., 2025). Early efforts such as BanglaBERT (Bhattacharjee et al., 2022), BanglaGPT (Salim et al., 2023), TituLLM (Nahin et al., 2025), and TigerLLM (Raihan and Zampieri, 2025) tried to address this challenge. Yet progress

¹<https://dipta007.github.io/GanitLLM/>

²[icls/most-spoken-languages-in-the-world](https://icls.org/most-spoken-languages-in-the-world)

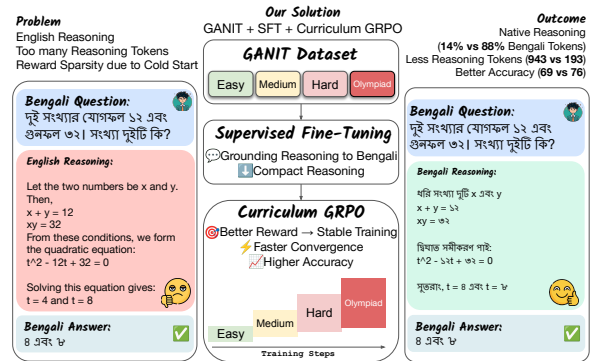


Figure 1: Overview of our approach for a Bengali mathematical reasoning model. (Left) Current models reason in English even for Bengali questions, resulting in reduced interpretability for native speakers. (Center) Our solution combines the GANIT dataset with SFT to ground reasoning in Bengali, followed by CURRICULUM-GRPO for efficient RL training. (Right) Our approach achieves native Bengali reasoning (88% Bengali vs. 14%), reduces reasoning tokens by 79%, and improves accuracy from 69 to 76.

in complex multi-step reasoning tasks in Bengali, particularly in mathematics, continues to lag due to the scarcity of high-quality Bengali mathematical reasoning datasets (Bhowmik et al., 2025). This underscores the need for research on Bengali mathematical reasoning to advance low-resource LLMs and expand their applications for Bengali speakers.

While previous works have shown promising results in Bengali mathematics (Lai and Nissim, 2024), they have only evaluated the final accuracy. Shi et al. (2023) found that intermediate reasoning in English leads to better reasoning performance (left in Fig. 1), but in our case, we want a model that not only answers correctly but also reasons in Bengali (right in Fig. 1) for end-user interpretability and understanding. Most mathematical LLM users, e.g., students, seek not only the answer but also the step-by-step reasoning to understand the solving process. Additionally, we identified that traditional RL training recipes, e.g., GRPO even with a high-

quality dataset fails to solve this problem due to the high dominance of high-resource languages in pre-training. Wu et al. (2025) have shown that only GRPO training is enough to improve the mathematics capability in Chinese but Chowdhery et al. (2022) revealed that the number of Bengali tokens in pre-training is ~ 15 times fewer than Chinese (0.026% vs 0.4%), which makes Bengali a far rarer and harder to improve than Chinese.

We define the cold-start problem in GRPO training as the scenario in which the policy model—due to its limited capability in the target low-resource language—fails to produce any correct solutions within a rollout group. This results in zero rewards across all samples and, consequently, zero gradients. Such cases lead to highly inefficient training (see §7.2), posing a critical challenge in low-resource settings. In this work, we introduce (i) **a difficulty-aware, rigorously-filtered and -processed high-quality Bengali math dataset, GANIT**, and (ii) **a novel training recipe, CURRICULUM-GRPO to tackle the cold-start problem in low-resource languages**. To the best of our knowledge, we are the first to develop a Bengali Math LLM that performs reasoning truly in Bengali, rather than translating (Shu et al., 2024) or reasoning in English (Lai and Nissim, 2024). To enable this, we construct a difficulty-tagged Bengali math dataset by combining and curating several existing high-quality Bengali math datasets and adopting the pass@k as the proxy difficulty score (§3). We categorize problems into Easy, Medium, Hard and Olympiad levels. Next, we fine-tune the base-instruct model on our CoT-SFT variant to teach the model to reason in Bengali rather than English. Finally, we apply Group Relative Policy Optimization (GRPO) (Shao et al., 2024) with three different reward scores (§4): (i) Format, (ii) Accuracy, (iii) Bengali Reasoning. We have also modified the dataset sampling during training to tackle the cold-start problem in Bengali reasoning. We call this whole reinforcement learning procedure CURRICULUM-GRPO (§4.1).

To summarize, our contributions are as follows:

1. We introduce a difficulty-tagged, rigorously-filtered Bengali Math dataset with verifiable answers.
2. We introduce a novel GRPO training recipe, **CURRICULUM-GRPO**, which can effectively tackle the cold-start problem of low-resource languages during group-based rewarding.
3. To the best of our knowledge, we release the **first Bengali Mathematical Reasoning model**. GANITLLM outperforms models that are twice its size and achieves performance comparable to models four times larger, all while reasoning natively in Bengali rather than relying on English and using 79.5% fewer tokens.

2 Related Works

Large language models (LLMs) have recently made significant progress in complex reasoning tasks (Du et al., 2025; Abdin et al., 2024) like mathematics (Wu et al., 2025; Zhang et al., 2025; Nazi et al., 2026), coding (Ceka et al., 2025; Halim et al., 2025; El-Kishky et al., 2025) and commonsense (Gawin et al., 2025; Wang et al., 2025b; Krause and Stolzenburg, 2024) reasoning. The OpenAI o1 model (OpenAI, 2024) achieves state-of-the-art reasoning on complex, multi-step tasks. However, OpenAI has never open-sourced their training recipe. DeepSeek-R1 (Guo et al., 2025) marks a major advance in reasoning through novel reinforcement learning-based training methods. These high-performance LLMs excel in mathematical reasoning for high-resource languages, but their limited interpretability in the target language highlights the need for multilingual research.

2.1 Multilingual Math Reasoning

There has been growing exploration of multilingual contexts. Lai and Nissim (2024) introduced mCoT, a 7B model with multilingual Chain-of-Thought tuning, achieving consistent reasoning across eleven languages on proprietary LLMs. Similarly, the MathCritique pipeline on ChatGLM3-32B enhances mathematical problem-solving while preserving language ability, outperforming larger LLMs (Xu et al., 2024). Confucius3-Math (Wu et al., 2025), a 14B open-source model for Chinese K-12 math, uses reinforcement learning with targeted entropy regularization to deliver state-of-the-art reasoning. Muennighoff et al. (2025) developed s1-32B with budget forcing, showing that simple test-time scaling yields up to 27% gains over o1-preview on competition math. MindMerger (Huang et al., 2024) integrates LLMs with multilingual models to augment cross-lingual reasoning, improving accuracy on MGSM (Shi et al., 2023) by up to 8% in low-resource settings without parameter updates. MathOctopus (Chen et al., 2023), trained on MGSM8K-Instruct with rejection sampling and parallel corpora, outperforms open-source models

Dataset	Problem	Solution	Source (if translated)	Size	Human Evaluation
mCoT-MATH-bn (Lai and Nissim, 2024)	GT	GT	mCoT-MATH (Lai and Nissim, 2024)	580k	100%
NuminaMath-CoT-bn (Rahman, 2024)	LT	LT	NuminaMath-CoT (LI et al., 2024)	859k	97%
s1k-Bangla (BanglaLLM, 2024)	LT	LT	s1K-1.1 (Muennighoff et al., 2025)	1k	96%
DL Sprint 3.0 (DL Sprint 3.0, 2024)	HA	HA	–	200	96%
SOMADHAN (Paul et al., 2025)	HT	HT	GSM8K (Cobbe et al., 2021)	8.7k	96%
Shomikoron (Aurpa et al., 2024)	HA	HA	–	3.4k	90%
Bangla-Math (kawchar Husain, 2024)	HA	LG	–	1.5k	88%
PatiGonit (Era et al., 2024)	HT	HT	MAWPS (Koncel-Kedziorski et al., 2016)	10k	85%
BMWP (Mondal et al., 2025)	HA	HA	–	8.6k	77%

HT = Human Translated, LT = LLM Translated, GT = Google Translated, HA = Human Annotated, and LG = LLM Generated

Table 1: Overview of the quality and statistics of open-source datasets based on manual evaluation. **We used only the datasets with human evaluation greater than 95% (top 5 rows).** Human Evaluation reports the percentage of samples with both correct problem statements and valid solutions out of the total sampled dataset.

and ChatGPT in few-shot multilingual math reasoning. While studies have explored multilingual settings or targeted specific languages (e.g. Chinese), low-resource languages such as Bengali remain underexplored. Moreover, most of the multilingual reasoning methods only target the final accuracy, while the reasoning tokens are still in English.

2.2 Bengali Math Reasoning

Bengali LLMs reasoning research is still in a rudimentary stage. Nahin et al. (2025) presented TituLLMs, the first large pretrained Bengali LLMs and five benchmarking datasets, underscoring the challenges of language adaptation. TigerLLM (Raihan and Zampieri, 2025) surpasses both open-source and proprietary models on standard benchmarks, setting a new baseline for Bengali reasoning. Research on Bengali mathematical reasoning has primarily concentrated on the development of high-quality datasets. Aurpa et al. (2024) and Era et al. (2024) introduced Shomikoron and PatiGonit, showing transformer models’ effectiveness on Bengali math problems. Mondal et al. (2025) contributed a large Bengali Math Word Problem (BMWP) dataset and showed strong operation prediction with deep neural networks. SOMADHAN (Paul et al., 2025), a manually created step-by-step reasoning dataset, illustrates that Chain-of-Thought prompting improves proprietary LLMs on multi-step tasks. BEnQA (Shafayat et al., 2024), a bilingual K-12 math dataset, indicates LLMs lag in Bengali math reasoning but improve when augmented with English translation prompts. Community translations of popular datasets like NuminaMath (LI et al., 2024) on Hugging Face and Kaggle broaden access, but ensuring cross-source alignment and quality remains challenging (Rahman, 2024; DL Sprint 3.0, 2024; BanglaLLM, 2024).

3 Creating GANIT

We construct GANIT, a rigorously-processed, difficulty-aware Bengali math dataset comprising both GANIT-TRAIN and GANIT-DEV sets. GANIT-TRAIN consists of two distinct splits: CoT-SFT variant and RLVR³, designed specifically for instruction tuning and reinforcement learning pipelines, respectively. Motivated by the limitations of existing Bengali evaluation benchmarks (details in §3.4), we additionally develop a hold-out, difficulty-aware dev set, GANIT-DEV, to assess the capabilities of GANITLLM. The entire dataset creation pipeline is illustrated in Fig. 2.

3.1 Data Collection

Prior research has shown that LLMs can achieve superior performance when trained on high-quality and diverse data, even when the overall data volume is limited (Muennighoff et al., 2025; Raihan et al., 2024). However, obtaining such high-quality datasets for low-resource languages remains challenging, particularly in specialized domains such as mathematics (Chen et al., 2023). To address this gap, we processed a large Bengali mathematical dataset (~1.5M) by collecting publicly available Bengali math datasets spanning human-authored, human-translated, LLM-translated, and Google-translated sources. The datasets span mathematical skills from basic arithmetic (Mondal et al., 2025) to advanced competition-level problems (DL Sprint 3.0, 2024). Additionally, they incorporate high-quality samples from research repositories (Paul et al., 2025) and community-contributed resources (Rahman, 2024). This comprehensive coverage enables us to get a seed dataset with good coverage across different genres of mathematics.

³Reinforcement Learning with Verifiable Rewards

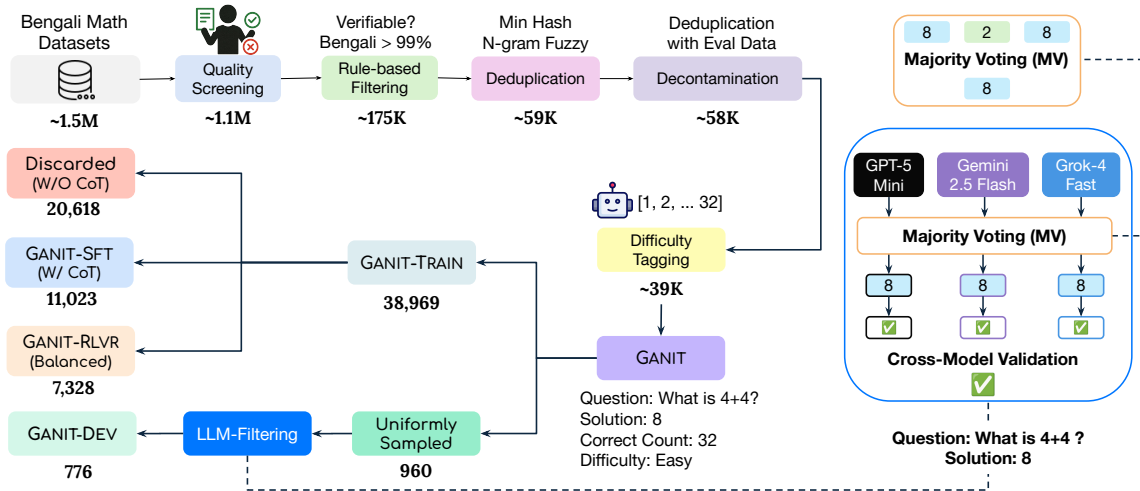


Figure 2: Overview of the GANIT construction pipeline. Starting from $\sim 1.5\text{M}$ Bengali math problems, we apply multi-stage quality filtration, verification, deduplication, and decontamination to obtain GANIT-TRAIN (SFT and RLVR) and GANIT-DEV.

3.2 Data Filtering

We applied a rigorous data filtering pipeline to the collected datasets listed in Table 1, ensuring that the resulting data is high-quality, well-formatted, deduplicated, and decontaminated.

Quality Screening: The problems and solutions in the datasets can be categorized as follows: (i) human-annotated, (ii) human-translated, (iii) LLM-generated, (iv) LLM-translated, and (v) Google-translated. Two evaluators manually evaluated randomly sampled subsets from all datasets (100 from each) to ensure rigorous quality screening. Manual evaluation results are reported in Table 1. The background and expertise of the evaluators are summarized in App. F.

As expected, the manual evaluation shows that human-annotated, human-translated, and LLM-translated datasets exhibit higher quality than LLM-generated synthetic datasets. After quality screening, only low-error datasets were retained (accuracy $> 95\%$), reducing the total size from $\sim 1.5\text{M}$ to $\sim 1.1\text{M}$ instances.

Rule-based Filtering: We applied rule-based filtering to the selected datasets to ensure consistency and verifiability. Specifically, (i) only solutions with numerical values were retained to allow for verifiable rewards, (ii) only problems containing at least 99% Bengali characters were included, and (iii) multiple-choice questions were excluded.

Deduplication: We employed a two-stage deduplication process: (i) fuzzy string-based matching using normalized Levenshtein distance to detect

near duplicates (3-gram with 70% threshold), and (ii) MinHash-based similarity detection (200 hash size with 50% threshold).

Decontamination: To prevent data leakage from evaluation benchmarks, we applied MinHash-based decontamination against MGSM (Shi et al., 2023) and MSVAMP (Chen et al., 2023). Training instances with similarity above 50% were removed. This ensured the training data was decontaminated, allowing reliable evaluation.

3.3 Difficulty-aware GANIT

Inspired by the pass@k (Chen et al., 2021) metric, we have used a similar strategy to estimate the difficulty level of each problem. First, we identify a strong general LLM based on its performance on Bn-MGSM (Shi et al., 2023) and Bn-MSVAMP (Chen et al., 2023) test sets. We choose these as they are the standard test sets for multilingual math evaluation. To that purpose, we have evaluated 8 open-source models ranging from 8B to 72B and identified that the Qwen3-32B performs the best (details in App. E). Next, we use Qwen3-32B to generate 32 independent solutions for each problem with a temperature of 0.7 to balance between diversity and correctness. A problem was retained only if the model successfully solved it at least once, thereby filtering out (i) potentially mislabeled noisy data, and (ii) instances likely unsolvable by smaller models during GRPO’s group rollout. Finally, we uniformly categorize problems into four difficulty buckets based on the number of successful generations: **Olympiad (1–8)**, **Hard (9–16)**, **Medium (17–24)**, and **Easy (25–32)**, enabling granular con-

Difficulty	MGSM	MSVAMP	GANIT-DEV
Easy	77.50	86.00	28.74
Medium	16.40	8.40	26.03
Hard	3.60	3.20	24.31
Olympiad	2.50	2.40	21.26

Table 2: Difficulty statistics (in % of total data) of the Bn-MGSM (Shi et al., 2023), Bn-MSVAMP (Chen et al., 2023) and GANIT-DEV datasets.

trol over complexity levels.

3.4 GANIT-DEV

Motivation: To assess the difficulty of the standard MGSM and MSVAMP datasets, we applied the same difficulty tagging pipeline to their Bengali counterparts, Bn-MGSM and Bn-MSVAMP. The fine-grained statistics are presented in Table 2. As the results indicate, the standard evaluation datasets in Bengali are relatively easy for current LLMs to solve. To construct a more robust difficulty-aware evaluation, we sampled 30 problems from each fine-grained difficulty bucket (1–32), resulting in a total of 960 examples (30×32).

LLM-based Filtering: Furthermore, to ensure the quality of the dev set, we applied an LLM-based filtering procedure using three proprietary models: **GPT-5-mini**, **Gemini-2.5-Flash**, and **Grok-4-Fast**. Each model was prompted to solve each problem independently three times. Following the majority voting strategy, we select an answer if it appears in at least two out of three generations. Then we mark the answer as correct or wrong for each of the models. Finally, we retain only those problems that were correctly solved by all three models. This dual-stage filtering process ensures that the final set contains only high-quality, validated problems, minimizing the risk of noisy data. Notably, even after both filtering stages, the distribution of examples across difficulty buckets remained relatively balanced, as shown in Table 2.

3.5 GANIT-TRAIN

In our training split, we specifically constructed two distinct datasets: (i) for instruction tuning (problem, CoT, solution), and (ii) for reinforcement learning (problem, verifiable answer). Since the instruction tuning data is primarily used to teach the LLM to reason in Bengali rather than to optimize for correctness, we hypothesize that SFT is less sensitive to data imbalance. Based on this

assumption, we constructed a difficulty-balanced dataset for the RL split and utilized the remaining (imbalanced) portion for instruction tuning.

To achieve fine-grained difficulty balancing, we moved beyond the standard four coarse buckets and instead considered the exact number of correct generations (ranging from 1 to 32). We then randomly sampled an equal number of instances for each count, ensuring uniform representation. This was particularly important for the RL split, which is more susceptible to overfitting. Full statistics of GANIT are provided in Table 9.

4 Training GANITLLM

Following the success of GRPO (Shao et al., 2024; Dipta et al., 2026) in training LLMs for reasoning, tool-calling, and math, we use GRPO to train our model. The total reward R is computed as:

$$R = R_{\text{format}} + R_{\text{correctness}} + R_{\text{bengali}} \in [0, 4]$$

where $R_{\text{format}} \in \{0, 1\}$ checks output format, $R_{\text{correctness}} \in \{0, 1, 2\}$ rewards correct answers (with bonus for Bengali answers), and $R_{\text{bengali}} \in \{0, 1\}$ rewards sufficient ($\geq 80\%$) Bengali reasoning. The 80% threshold is intentionally set below 100% to ensure that sufficient token budget remains for mathematical notation and formulas, which are inherently language-agnostic. This threshold was empirically determined based on the distribution of Bengali tokens in the chain-of-thought traces available in our GANIT-SFT dataset. Furthermore, during reward computation, we strip punctuation from the token count to avoid artificially inflating or skewing the Bengali ratio score. From initial runs, we had the following observations:

1. The policy model, i.e., Qwen3, tends to reason in English and then produce the Bengali answer, even when explicitly prompted to reason and answer in Bengali (App. G). We hypothesize that this behavior stems from the predominance of English reasoning data during pre-training.
2. Under standard shuffle-based GRPO training, the policy model fails to generate any correct answers within the rollout group, causing all advantage values to collapse to zero. As a result, the model fails to learn effectively.
3. Many of the early generations are truncated due to the maximum token limit we imposed. These truncated outputs negatively impact the learning

process. While increasing the token limit is possible, we observed that longer generations often contain repetition or unnecessary reasoning.

Building on these observations, we introduce a multi-stage training recipe. **In the first stage**, we leverage the CoT-SFT split of GANIT-TRAIN to teach the model to reason in Bengali using fewer tokens. **In the second stage**, we apply a modified GRPO training procedure (described in §4.1) using the RL split of GANIT-TRAIN, enabling the model to generalize its reasoning ability and effectively solve Bengali math problems. To further mitigate the challenges of overlength generations, we follow Yu et al. (2025) and incorporate an overlength filter and token-level loss into the GRPO training.

4.1 CURRICULUM-GRPO

As discussed earlier, using a traditional training strategy with random shuffling can result in “Hard” or “Olympiad” level problems appearing early in the training process – before the model has developed the ability to solve them, even with a high rollout of 8. This is expected, as the hard or olympiad problems are those that even the large models (e.g. Qwen3-32B) take 32 turns to get an accurate solution. In such cases, the model receives zero reward across the whole group, leading to ineffective updates and stagnation in learning.

To address this, and inspired by recent advances in curriculum-based learning (Hammoud et al., 2025; Chen et al., 2025; Gao et al., 2025), we propose **CURRICULUM-GRPO**, a modified data sampling strategy that orders training data based on pseudo-difficulty.

A naive approach would be to sort the entire dataset from easy to hard based on difficulty. However, this can lead to early overfitting on simpler problems, making it harder for the model to adapt to more challenging samples later in training. To mitigate this, we adopt a soft curriculum strategy, using the number of correct generations as a fine-grained difficulty signal (ranging from 1 to 32) rather than relying on coarse difficulty categories. This provides more precise control over difficulty-aware sampling and allows the model to gradually strengthen its reasoning capabilities through increasingly difficult examples.

Specifically, for every bucket (1–32), we sample 60% of examples (136 instances) from the current bucket and 40% from the remaining 31 buckets (3 instances per bucket, totaling 93), resulting in

Algorithm 1 Curriculum-based dataset shuffling. Given a dataset \mathcal{D} , the algorithm outputs a curriculum-ordered version \mathcal{D}' , where each datapoint is tagged with a correctness count ranging from 1 (hardest) to 32 (easiest).

Require: Dataset $\mathcal{D} = \{(x_i, y_i, d_i)\}_{i=1}^N$ where $d_i \in [1, 32]$ is difficulty

Ensure: Curriculum-ordered dataset \mathcal{D}'

```

1: Group samples by difficulty:  $\mathcal{D}_d \leftarrow \{(x_i, y_i) : d_i = d\}$  for  $d = 1, \dots, 32$ 
2:  $n_d \leftarrow |\mathcal{D}_d|$   $\triangleright$  Total Samples per difficulty
3:  $n_p \leftarrow \lfloor n_d \times 0.6 \rfloor$   $\triangleright$  60% Primary samples
4:  $n_o \leftarrow \lfloor n_d \times 0.4 \rfloor$   $\triangleright$  40% Other Samples
5:  $\mathcal{D}' \leftarrow \emptyset$ 
6: for  $d_p = 32$  down to 1 do  $\triangleright$  Easy to Hard
7:    $\mathcal{B} \leftarrow \mathcal{D}_{d_p}[1 : n_p]$   $\triangleright$  Primary samples
8:   for each  $d_o \neq d_p$  do
9:      $\mathcal{B} \leftarrow \mathcal{B} \cup \mathcal{D}_{d_o}[\text{slice}]$   $\triangleright$  Add  $n_o$  samples
10:  end for
11:  Shuffle  $\mathcal{B}$  randomly
12:   $\mathcal{D}' \leftarrow \mathcal{D}' \cup \mathcal{B}$ 
13: end for
14: return  $\mathcal{D}'$ 

```

229 examples per bucket. We chose a 60/40 split based on preliminary experiments showing that higher primary-bucket ratios (e.g., 80/20) led to catastrophic forgetting of easier problems, while lower ratios (e.g., 50/50) diluted the curriculum signal. The 60/40 balance empirically provided stable training while maintaining sufficient difficulty progression. Finally, we sort the training data by the primary bucket’s difficulty level to ensure a smooth curriculum progression from easy to hard, reducing the risk of premature convergence or reward sparsity. The full procedure is detailed in Alg. 1.

5 Experiment Setup

Datasets For SFT and GRPO training, we use GANIT-SFT and GANIT-RLVR, respectively. For evaluation, we use Bn-MGSM (Shi et al., 2023) and Bn-MSVAMP (Chen et al., 2023) benchmarks. Additionally, we report performance on our proposed GANIT-DEV set.

Implementation Details We trained our model for 50 epochs during SFT and 5 epochs during GRPO. In both stages, the best checkpoint was selected based on accuracy on GANIT-DEV. We used full fine-tuning for SFT and LoRA-based fine-tuning (Hu et al., 2022) for GRPO (Shao et al.,

2024). All training and inference were conducted on 2×A100 GPUs. As base models, we used Qwen3-0.6B, Qwen3-1.7B, and Qwen3-4B (Yang et al., 2025). Additional implementation details are provided in App. D. For evaluation, we used a temperature of 0.0 to ensure deterministic reproducibility, and applied the same prompt across all models (see App. G).

Baselines While the primary goal of our paper is to develop a small Math LLM suitable for developing and low-resource countries, we also provide comparisons with larger models to compare our results. Specifically, we compare against the Qwen3 family, ranging from 0.6B to 32B parameters.⁴ For broader context, we also include results from gpt-4.1, gpt-4.1-mini, and TigerLLM-9B.

6 Results & Analysis

The results on MGSM and MSVAMP are reported in Table 3. In addition to accuracy, we report the number of generated words and the percentage of Bengali characters. Our goal is not only to develop a Bengali math LLM that produces correct answers, but also one that reasons in Bengali while maintaining a reasonable output length. This ensures both interpretability and practical usability for end users with limited computational resources. Due to significant differences in tokenization between English and Bengali, we report word counts rather than token counts for fair comparison. To compute Bengali reasoning percentage, we count all characters in the model’s reasoning output (excluding whitespace) and calculate what fraction belong to the Bengali Unicode block (U+0980–U+09FF). For instance, if a reasoning block contains 800 Bengali characters and 200 English/numeric characters, the Bengali percentage is 80%.

The results indicate that while base models perform reasonably well, their success often comes at the cost of longer generations and reasoning primarily in English. This is expected, as much of the reasoning-related pretraining data is in English (Chowdhery et al., 2022), often with long reasoning traces (Guo et al., 2025).

Training the model with GANIT-SFT improves both accuracy and interpretability, yielding better performance, fewer generated words, and reason-

⁴We focus on a single model family to isolate the effect of our training recipe from base model differences. Additionally, no existing Bengali math reasoning models are publicly available for direct comparison.

	MGSM ↑	MSVAMP ↑	Words ↓	Bn (%) ↑
gpt4.1	89.20	82.30	200	88.16
gpt4.1-mini	87.20	78.60	232	88.18
TigerLLM-9B	47.20	40.40	206	93.69
Qwen3-32B	85.60	76.10	712	21.08
Qwen3-14B	83.60	75.80	767	17.87
Qwen3-8B	75.12	72.42	846	16.48
Qwen3-4B	69.20	70.50	943	14.79
GANITLLM-4B	76.80	76.40	193	88.71
Qwen3-1.7B	15.20	14.10	1124	19.64
GANITLLM-1.7B	52.80	66.80	210	87.80
Qwen3-0.6B	8.40	12.20	1265	12.43
GANITLLM-0.6B	28.40	52.40	248	88.70

Table 3: Results on Bn-MGSM and Bn-MSVAMP test sets. **GANITLLM enables smaller models to match or exceed larger counterparts:** GANITLLM-4B surpasses Qwen3-8B by 7.6 points on MGSM while improving Bengali characters from 14.79% to 88.71%. **Bold** denotes best performance within each parameter category. Full results are provided in Table 7.

ing in Bengali. Further improvements are observed with CURRICULUM-GRPO, which pushes the results even further in the desired direction.

Notably, our 4B model outperforms the 8B base model and shows comparable results with the 14B model while being 2x and 3.5x smaller. Beyond accuracy, our approach yields dramatically more concise responses, GANITLLM-4B generates answers averaging only 193 words compared to 943 for the base model, **a 79.5% reduction**. More importantly, **Bengali characters increase from 14.79% to 88.71%**, indicating that our models reason in the target language rather than defaulting to English, addressing a critical limitation of Mathematical Reasoning. While improvements remain consistent, the absolute gains from CURRICULUM-GRPO narrow at the 0.6B scale, suggesting that model capacity imposes a floor on achievable performance for complex mathematical reasoning. Also, previous studies (Nimmaturi et al., 2025) have shown that a sufficiently capable model is needed to improve on reasoning with GRPO. We also provide a qualitative analysis in §9.

7 Ablation Study

7.1 Impact of Multi-stage Training

As the base-instruct models already show strong instruction-following capabilities, an intuitive question arises: *Do we really need multi-stage training? Or could we do the SFT/RL stage directly on top of the base-instruct model?* Table 4 presents an ablation study showing the necessity of our multi-

	MGSM \uparrow	MSVAMP \uparrow	Words \downarrow	Bn (%) \uparrow
Qwen3-4B	69.20	70.50	943	14.79
+ SFT	74.00	74.60	184	86.65
+ CGRPO	82.40	78.50	844	14.94
GANITLLM-4B	76.80	76.40	193	88.71
Qwen3-1.7B	15.20	14.10	1124	19.64
+ SFT	48.80	64.60	253	87.79
+ CGRPO	59.60	66.20	1002	18.74
GANITLLM-1.7B	52.80	66.80	210	87.80
Qwen3-0.6B	8.40	12.20	1265	12.43
+ SFT	28.40	51.40	263	88.60
+ CGRPO	17.20	35.20	824	11.67
GANITLLM-0.6B	28.40	52.40	248	88.70

Table 4: Ablation study on multi-stage training. CURRICULUM-GRPO alone achieves competitive accuracy but fails to maintain Bengali language reasoning, while SFT alone provides limited reasoning improvements. **Our multi-stage pipeline combines the complementary benefits of both approaches.**

stage training pipeline. We compare three training configurations: SFT only, CURRICULUM-GRPO (CGRPO) only, and our full pipeline (GANITLLM = SFT followed by CGRPO).

SFT establishes language grounding but provides limited reasoning gains. SFT alone increases Bengali character usage (from 14.79% to 86.65% for the 4B model) while reducing the number of tokens. However, accuracy improvements are modest compared to CGRPO-based training.

CURRICULUM-GRPO alone improves accuracy but sacrifices interpretability. Applying CGRPO directly to the base model yields the highest raw accuracy, occasionally surpassing the 14B counterpart. However, this configuration retains only 14.94% Bengali in reasoning, indicating that the model primarily reasons in English. This defeats the purpose of developing a Bengali mathematical reasoning system, as users receive little interpretable reasoning in their native language.

Multi-stage training achieves the best trade-off. Our sequential approach: first grounding the reasoning in Bengali through SFT, then enhancing reasoning via CGRPO—yields strong accuracy while maintaining high language adherence (88.71% Bengali characters) and concise outputs (193 words). This demonstrates that the two stages serve complementary roles that cannot be achieved through single-stage training alone.

7.2 Impact of CURRICULUM-GRPO

Table 5 presents an ablation comparing standard GRPO against our proposed CURRICULUM-GRPO (CGRPO) within the multi-stage training

	MGSM \uparrow	MSVAMP \uparrow	Words \downarrow	Bn (%) \uparrow	Best Ckpt. \downarrow
Qwen3-4B	69.20	70.50	943	14.79	-
SFT + GRPO	77.60	76.30	189	88.61	2300
SFT + CGRPO	76.80	76.40	193	88.71	600
Qwen3-1.7B	15.20	14.10	1124	19.64	-
SFT + GRPO	53.60	66.90	207	88.32	7900
SFT + CGRPO	52.80	66.80	210	87.80	2100
Qwen3-0.6B	8.40	12.20	1265	12.43	-
SFT + GRPO	32.40	52.50	246	88.45	7300
SFT + CGRPO	28.40	52.40	248	88.70	1300

Table 5: Ablation study comparing GRPO and CGRPO. Both methods achieve comparable accuracy, but **CGRPO reaches optimal performance 3.8-5.6 \times faster** by addressing the cold start problem through curriculum-based sample ordering. Best Checkpoint denotes the training step at which peak validation performance was achieved.

pipeline. While both approaches achieve comparable final performance, CGRPO demonstrates substantially improved training efficiency.

Comparable accuracy with dramatically faster convergence. Across all model scales, GRPO and CGRPO achieve nearly identical accuracy on both benchmarks. For the 4B model, the difference is within 1 percentage point (77.60 vs. 76.80 on MGSM; 76.30 vs. 76.40 on MSVAMP). However, CGRPO reaches its optimal checkpoint at step 600 compared to step 2300 for vanilla GRPO – a 3.8 \times reduction in training steps. This efficiency gap widens at smaller scales: for the 0.6B model, CGRPO converges at step 1300 versus 7300 for GRPO, representing a 5.6 \times speedup.

Addressing the cold start problem. The efficiency gains stem from CGRPO’s curriculum-based sample ordering. In vanilla GRPO, random shuffling exposes the model to difficult examples early in training when it lacks sufficient capability, resulting in predominantly incorrect generations that provide weak or no learning signals at all (i.e. all rewards 0 across the group). This cold start problem delays meaningful policy improvement. On the other hand, CGRPO orders training samples by difficulty, allowing the model to first build foundational reasoning patterns before tackling complex examples.

8 Cold-Start Problem during GRPO

We define the cold-start problem in GRPO as the scenario in which the policy model fails to generate any correct solution across the entire rollout group, resulting in zero gradients and suboptimal training. In Fig. 3, we plot checkpoint-wise accuracy on both MGSM and MSVAMP datasets under two

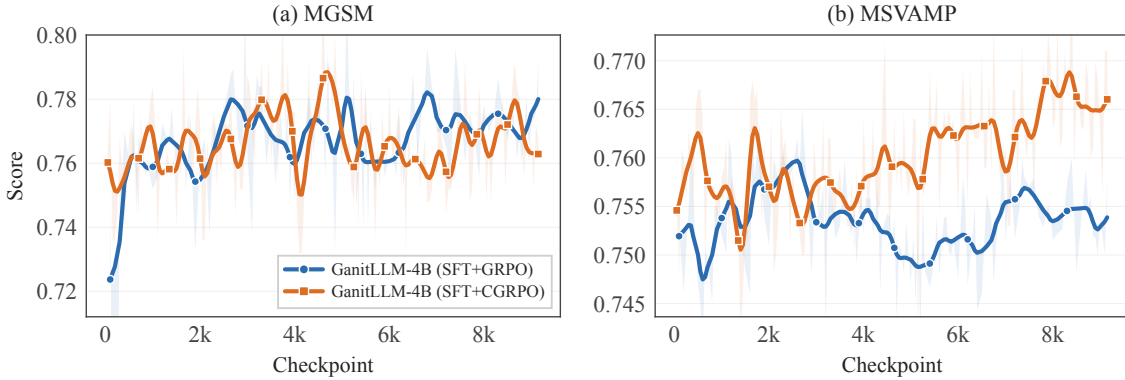


Figure 3: Evaluation curves comparing GRPO and CURRICULUM-GRPO on MGSM and MSVAMP benchmarks. Checkpoint-wise accuracy demonstrates that while both methods achieve comparable performance on the easier MGSM dataset (left), CGRPO substantially outperforms traditional GRPO on the harder MSVAMP dataset (right), where the cold-start problem causes GRPO to stagnate.

configurations: (i) SFT \rightarrow GRPO and (ii) SFT \rightarrow CURRICULUM-GRPO.

The results show that on MGSM, the easier of the two datasets, traditional GRPO struggles initially but eventually catches up in accuracy after several hundred steps. In contrast, on MSVAMP, our proposed CURRICULUM-GRPO method demonstrates more efficient learning, exhibiting a steady upward trend in accuracy. Meanwhile, traditional GRPO stagnates, likely due to local optima caused by zero-gradient updates.

9 Qualitative Analysis

In Fig. 4, we present model outputs for a representative Olympiad-level problem to illustrate the differences between training configurations. Results show that the base model (Qwen3-4B) produces correct answers but reasons primarily in English (7.58% Bengali) with verbose outputs (932 words). SFT alone achieves high Bengali usage (97.63%) and conciseness (645 words) but fails to produce the correct answer. CURRICULUM-GRPO alone improves accuracy but maintains English reasoning (7.32% Bengali) and generates extremely verbose outputs (2223 words). Our full pipeline (SFT \rightarrow CURRICULUM-GRPO) combines the benefits of both stages: achieving correct answers, native Bengali reasoning (97.7%), and concise outputs (467 words), demonstrating that the two training stages serve complementary roles.

Key Findings

1. **SFT grounds language but not reasoning ability.** SFT alone successfully shifts the model’s reasoning from English to Bengali (8% \rightarrow 97% Bengali tokens) and reduces

verbosity (932 \rightarrow 645 words). However, it doesn’t generate the accurate solution, suggesting that language grounding and mathematical reasoning are orthogonal capabilities.

2. **GRPO improves accuracy but not language.** GRPO alone improves accuracy on difficult problems but the model continues to reason in English (8% Bengali tokens).
3. **Two-stage training combines both benefits.** Our SFT \rightarrow GRPO pipeline produces outputs that are simultaneously accurate (matching GRPO-only), Bengali-dominant (matching SFT-only), and concise. This validates our hypothesis that SFT provides the language foundation that GRPO can then optimize without losing the language fidelity.

10 Conclusion

We address a critical gap in multilingual math reasoning: even when a base model can solve Bengali problems, it often reasons in English and merely translates the final answer. Our extensive ablations further show that traditional GRPO alone is inefficient for effective math reasoning in low-resource settings. To overcome these challenges, we first introduce GANIT, a comprehensive, difficulty-aware Bengali math corpus with three splits: CoT-SFT, RLVR, and a validation set. Building on this resource, we then propose CURRICULUM-GRPO, a novel data-sampling strategy that significantly improves the efficiency of GRPO training for low-resource and underrepresented languages. Experiments show that our approach outperforms strong baselines in accuracy, token efficiency, and language fidelity, while converging faster than standard GRPO.

Limitations

While our method introduces a novel training paradigm to address the cold-start problem in low-resource settings, several limitations remain.

First, our study is limited to a single language (Bengali) and a single domain (mathematical word problems). How well the proposed data construction and training recipe transfer to other low-resource languages remains an open research question.

Second, several components of our pipeline rely on proxy signals and automated tools. For instance, we tier difficulty using Pass@k scores from an evaluator model and filter development data using strong LLMs. Both can potentially introduce model-specific biases (e.g., systematically mischaracterizing certain problem types) that may propagate into training decisions.

Finally, our language-fidelity reward employs a character-percentage heuristic to approximate “Bengali reasoning,” which may incorrectly penalize valid outputs that mix languages, contain transliteration, or use symbols and numerals in ways that correlate with problem difficulty – both of which the heuristic cannot distinguish from true code-switching.

Despite these limitations, we believe our work offers a strong first step toward grounded, language-consistent reasoning in low-resource settings and provides a practical, extensible training framework that can be adapted and refined in future research.

Acknowledgment

Some experiments were conducted on the UMBC HPCF, supported by the National Science Foundation under Grant No. CNS-1920079. This material is also based on research that is in part supported by DARPA for the SciFy program under agreement number HR00112520301. The U.S. Government is authorized to reproduce and distribute reprints for Governmental purposes notwithstanding any copyright notation thereon. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies or endorsements, either express or implied, of DARPA or the U.S. Government.

References

Marah Abdin, Jyoti Aneja, Harkirat Behl, Sébastien Bubeck, Ronen Eldan, Suriya Gunasekar, Michael

Harrison, Russell J Hewett, Mojan Javaheripi, Piero Kauffmann, and 1 others. 2024. [Phi-4 technical report](#). *ArXiv preprint*, abs/2412.08905.

Tanjim Taharat Aurpa, Kazi Noshin Fariha, and Kawser Hossain. 2024. Shomikoron: Dataset to discover equations from bangla mathematical text. *Data in Brief*, 55:110742.

BanglaLLM. 2024. [s1k-32-bangla](#). Hugging Face dataset.

Abhik Bhattacharjee, Tahmid Hasan, Wasi Ahmad, Kazi Samin Mubasshir, Md Saiful Islam, Anindya Iqbal, M. Sohel Rahman, and Rifat Shahriyar. 2022. [BanglaBERT: Language model pretraining and benchmarks for low-resource language understanding evaluation in Bangla](#). In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1318–1327, Seattle, United States. Association for Computational Linguistics.

Shimanto Bhowmik, Tawsif Tashwar Dipto, Md Sazzad Islam, Sheryl Hsu, and Tahsin Reasat. 2025. [Evaluating llms’ multilingual capabilities for bengali: Benchmark creation and performance analysis](#). *ArXiv preprint*, abs/2507.23248.

Ira Ceka, Saurabh Pujar, Irene Manotas, Gail Kaiser, Baishakhi Ray, and Shyam Ramji. 2025. [How does llm reasoning work for code? a survey and a call to action](#). *ArXiv preprint*, abs/2506.13932.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and 1 others. 2021. [Evaluating large language models trained on code](#). *ArXiv preprint*, abs/2107.03374.

Nuo Chen, Zinan Zheng, Ning Wu, Ming Gong, Dongmei Zhang, and Jia Li. 2023. [Breaking language barriers in multilingual mathematical reasoning: Insights and observations](#). *ArXiv preprint*, abs/2310.20246.

Xiaoyin Chen, Jiarui Lu, Minsu Kim, Dinghuai Zhang, Jian Tang, Alexandre Piché, Nicolas Gontier, Yoshua Bengio, and Ehsan Kamaloo. 2025. [Self-Evolving Curriculum for LLM Reasoning](#).

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, and 48 others. 2022. [PaLM: Scaling Language Modeling with Pathways](#).

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark 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*, abs/2110.14168.

- Shubhashis Roy Dipta, Daniel Bis, Kun Zhou, Lichao Wang, Benjamin Z Yao, Chenlei Guo, and Ruhi Sarikaya. 2026. Pa3: Policy-aware agent alignment through chain-of-thought. *arXiv preprint arXiv:2603.14602*.
- DL Sprint 3.0. 2024. [DL Sprint 3.0: Bengali AI Math Olympiad](#). Kaggle competition.
- Dong Du, Shulin Liu, Tao Yang, Shaohua Chen, and Yang Li. 2025. [Ulurl: An ultra-long output reinforcement learning approach for advancing large language models' reasoning abilities](#). *ArXiv preprint*, abs/2507.19766.
- Ahmed El-Kishky, Alexander Wei, Andre Saraiva, Borys Minaiev, Daniel Selsam, David Dohan, Francis Song, Hunter Lightman, Ignasi Clavera, Jakub Pachocki, and 1 others. 2025. [Competitive programming with large reasoning models](#). *ArXiv preprint*, abs/2502.06807.
- Jalisha Jashim Era, Bidyarthi Paul, Tahmid Sattar Aothoi, Mirazur Rahman Zim, and Faisal Muhammad Shah. 2024. Empowering bengali education with ai: Solving bengali math word problems through transformer models. In *2024 27th International Conference on Computer and Information Technology (ICCIT)*, pages 909–914. IEEE.
- Zhaolin Gao, Joongwon Kim, Wen Sun, Thorsten Joachims, Sid Wang, Richard Yuanzhe Pang, and Liang Tan. 2025. [Prompt Curriculum Learning for Efficient LLM Post-Training](#).
- Cole Gawin, Yidan Sun, and Mayank Kejriwal. 2025. Navigating semantic relations: Challenges for language models in abstract common-sense reasoning. In *Companion Proceedings of the ACM on Web Conference 2025*, pages 971–975.
- Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, and 1 others. 2025. [Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning](#). *ArXiv preprint*, abs/2501.12948.
- Kevin Halim, Sin G Teo, Ruitao Feng, Zhenpeng Chen, Yang Gu, Chong Wang, and Yang Liu. 2025. [A study on thinking patterns of large reasoning models in code generation](#). *ArXiv preprint*, abs/2509.13758.
- Hasan Abed Al Kader Hammoud, Kumail Alhamoud, Abed Hammoud, Elie Bou-Zeid, Marzyeh Ghassemi, and Bernard Ghanem. 2025. [Train Long, Think Short: Curriculum Learning for Efficient Reasoning](#).
- Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. [Lora: Low-rank adaptation of large language models](#). In *The Tenth International Conference on Learning Representations, ICLR 2022, Virtual Event, April 25-29, 2022*. OpenReview.net.
- Zixian Huang, Wenhao Zhu, Gong Cheng, Lei Li, and Fei Yuan. 2024. [Mindmerger: Efficiently boosting LLM reasoning in non-english languages](#). In *Advances in Neural Information Processing Systems 38: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.
- kawchar Husain. 2024. [Bangla-Math](#). Hugging Face dataset.
- Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. 2016. [MAWPS: A math word problem repository](#). In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1152–1157, San Diego, California. Association for Computational Linguistics.
- Stefanie Krause and Frieder Stolzenburg. 2024. [From data to commonsense reasoning: the use of large language models for explainable ai](#). *ArXiv preprint*, abs/2407.03778.
- Huiyuan Lai and Malvina Nissim. 2024. [mcot: Multilingual instruction tuning for reasoning consistency in language models](#). *ArXiv preprint*, abs/2406.02301.
- Jia LI, Edward Beeching, Lewis Tunstall, Ben Lipkin, Roman Soletskyi, Shengyi Costa Huang, Kashif Rasul, Longhui Yu, Albert Jiang, Ziju Shen, Zihan Qin, Bin Dong, Li Zhou, Yann Fleureau, Guillaume Lample, and Stanislas Polu. 2024. [Numinamath](#).
- Sanchita Mondal, Debnarayan Khatua, Sourav Mandal, Dilip K Prasad, and Arif Ahmed Sekh. 2025. [BMWPs: the first Bengali math word problems dataset for operation prediction and solving](#). *Discover Artificial Intelligence*, 5(1):1–15.
- Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. 2025. [s1: Simple test-time scaling](#). *ArXiv preprint*, abs/2501.19393.
- Shahriar Kabir Nahin, Rabindra Nath Nandi, Sagor Sarker, Quazi Sarwar Muhtaseem, Md Kowsher, Apu Chandraw Shill, Md Ibrahim, Mehadi Hasan Menon, Tareq Al Muntasir, and Firoj Alam. 2025. [Titullms: A family of bangla llms with comprehensive benchmarking](#). *ArXiv preprint*, abs/2502.11187.
- Zabir Al Nazi, Shubhashis Roy Dipta, and Sudipta Kar. 2026. [†dagger: Distractor-aware graph generation for executable reasoning in math problems](#). *arXiv preprint arXiv:2601.06853*.
- Datta Nimmatouri, Vaishnavi Bhargava, Rajat Ghosh, Johnu George, and Debojyoti Dutta. 2025. [Predictive Scaling Laws for Efficient GRPO Training of Large Reasoning Models](#).
- OpenAI. 2024. [OpenAI. learning to reason with llms](#). [Open AI blog, 2024](#). *OpenAI Blog*.

- Bidyarthi Paul, Jalisha Jashim Era, Mirazur Rahman Zim, Tahmid Sattar Aothoi, and Faisal Muhammad Shah. 2025. [Leveraging large language models for bengali math word problem solving with chain of thought reasoning](#). *ArXiv preprint*, abs/2505.21354.
- Ashrafur Rahman. 2024. [NuminaMath-CoT-bn](#). Kaggle dataset.
- Nishat Raihan, Joanna Santos, and Marcos Zampieri. 2024. [Mojobench: Language modeling and benchmarks for mojo](#). *ArXiv preprint*, abs/2410.17736.
- Nishat Raihan and Marcos Zampieri. 2025. [Tigerllm-a family of bangla large language models](#). *ArXiv preprint*, abs/2503.10995.
- Md Shahidul Salim, Hasan Murad, Dola Das, and Faisal Ahmed. 2023. [Banglagpt: A generative pretrained transformer-based model for bangla language](#). In *2023 International Conference on Information and Communication Technology for Sustainable Development (ICICT4SD)*, pages 56–59. IEEE.
- Sheikh Shafayat, H Hasan, Minhajur Mahim, Rifki Putri, James Thorne, and Alice Oh. 2024. [BEnQA: A question answering benchmark for Bengali and English](#). In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1158–1177, Bangkok, Thailand. Association for Computational Linguistics.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. [Deepseekmath: Pushing the limits of mathematical reasoning in open language models](#). *ArXiv preprint*, abs/2402.03300.
- Freda Shi, Mirac Suzgun, Markus Freitag, Xuezhi Wang, Suraj Srivats, Soroush Vosoughi, Hyung Won Chung, Yi Tay, Sebastian Ruder, Denny Zhou, Dipanjan Das, and Jason Wei. 2023. [Language models are multilingual chain-of-thought reasoners](#). In *The Eleventh International Conference on Learning Representations, ICLR 2023, Kigali, Rwanda, May 1-5, 2023*. OpenReview.net.
- Peng Shu, Junhao Chen, Zhengliang Liu, Hui Wang, Zihao Wu, Tianyang Zhong, Yiwei Li, Huaqin Zhao, Hanqi Jiang, Yi Pan, Yifan Zhou, Constance Owl, Xiaoming Zhai, Ninghao Liu, Claudio Saunt, and Tianming Liu. 2024. [Transcending Language Boundaries: Harnessing LLMs for Low-Resource Language Translation](#).
- Guijin Son, Jiwoo Hong, Hyunwoo Ko, and James Thorne. 2025. [Linguistic Generalizability of Test-Time Scaling in Mathematical Reasoning](#).
- Yiming Wang, Pei Zhang, Jialong Tang, Haoran Wei, Baosong Yang, Rui Wang, Chenshu Sun, Feitong Sun, Jiran Zhang, Junxuan Wu, and 1 others. 2025a. [Polymath: Evaluating mathematical reasoning in multilingual contexts](#). *ArXiv preprint*, abs/2504.18428.
- Yiqun Wang, Sile Hu, Chaoqun Wan, Yonggang Zhang, Xiang Tian, Yaowu Chen, Xu Shen, and Jieping Ye. 2025b. [Deciphering and enhancing commonsense reasoning in LLMs from the perspective of intrinsic factual knowledge retrieval](#).
- Lixin Wu, Na Cai, Qiao Cheng, Jiachen Wang, and Yitao Duan. 2025. [Confucius3-Math: A Lightweight High-Performance Reasoning LLM for Chinese K-12 Mathematics Learning](#). *ArXiv preprint*, abs/2506.18330.
- Yifan Xu, Xiao Liu, Xinghan Liu, Zhenyu Hou, Yueyan Li, Xiaohan Zhang, Zihan Wang, Aohan Zeng, Zhengxiao Du, Wenyi Zhao, and 1 others. 2024. [Chatglm-math: Improving math problem-solving in large language models with a self-critique pipeline](#). *ArXiv preprint*, abs/2404.02893.
- An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. [Qwen3 Technical Report](#).
- Qiyang Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, and 1 others. 2025. [Dapo: An open-source llm reinforcement learning system at scale](#). *ArXiv preprint*, abs/2503.14476.
- Zhenru Zhang, Chujie Zheng, Yangzhen Wu, Beichen Zhang, Runji Lin, Bowen Yu, Dayiheng Liu, Jingren Zhou, and Junyang Lin. 2025. [The lessons of developing process reward models in mathematical reasoning](#). *ArXiv preprint*, abs/2501.07301.

Appendix

A Code-Switching Analysis of GANITLLM

To quantify the extent of code-switching in our model’s outputs, we conduct a fine-grained token-level analysis across all 1,236 GANITLLM-4B generations on Bn-MGSM and Bn-MSVAMP. We classify each non-Bengali token into one of five categories and compute the code-switch rate as the number of Bengali↔English word-level transitions per sentence, excluding mathematical notation.

As shown in Table 6, only 115 out of 101,869 tokens (0.11%) are actual English words, and these are predominantly domain-specific nouns carried over from the problem statements (e.g., “dvd,” “tv,” “iphone”) rather than instances of reasoning in English. With an average of just 0.014 code-switches per sentence, virtually every sentence produced by the model is monolingual Bengali. The non-Bengali content is overwhelmingly composed of mathematical notation and numerals, which are standard conventions even in human-written Bengali mathematics and do not constitute code-switching in any meaningful linguistic sense.

B Additional Results

Full results across MGSM, MSVAMP and GANIT-DEV are reported in Table 7.

We have also used two out-of-domain datasets: PolyMath (Wang et al., 2025a) and MT-MATH100 (Son et al., 2025). The results in Table 8 show that performance drops as difficulty increases (80.0% → 17.76% across PolyMath tiers). We attribute this to two factors: (1) our training data targets grade-level reasoning, lacking harder problem distributions, and (2) GANITLLM was trained to output direct numerical answers, whereas problems in PolyMath and MT-MATH100 often require equation-level or symbolic outputs. Both limitations call for more challenging, diversified training data, which we identify as future work.

Nevertheless, three findings support our core contribution of efficient GRPO training for low-resource languages:

- GANITLLM-4B-SFT+CGRPO (80.0%) surpasses Qwen3-8B (70.4%) and approaches Qwen3-14B (82.4%) on PolyMath-low, matching models 2–4× its size.
- Bengali adherence is consistently high (75–87%) across all difficulty levels, while

Qwen3-4B drops to 19–24% Bengali on harder tiers, essentially reasoning in English.

- GRPO improves over SFT even on harder benchmarks (+11.87 on MT-MATH100), validating our core contribution of effective Curriculum-GRPO training for low-resource language reasoning.

C GANIT Statistics

Table 9 shows the difficulty and source distribution of our proposed dataset, GANIT.

D Implementation Details

Full details of the hyperparameters used in the SFT and GRPO training is provided in the Tables 10 and 11, respectively.

E Evaluating Bn-MGSM & Bn-MSVAMP

Table 12 presents the zero-shot evaluation results of eight open-source LLMs on the Bengali splits of both datasets. Rather than using single-point accuracy, we compute Pass@4, as it aligns with our difficulty tagging procedure based on Pass@k. Based on these results, we select Qwen3-32B as the evaluator model in our difficulty tagging pipeline.

F Evaluator Details

Two of the authors conducted quality screening of the seed data. Both hold graduate-level degrees, ensuring strong domain expertise and analytical skills. Their native language is Bengali, and they also possess advanced proficiency in English, enabling accurate evaluation of bilingual and technical content. Additionally, their advanced mathematical background qualifies them to assess materials requiring precise reasoning and quantitative understanding.

G User Prompt

We use the following user prompt templates for all of our experiments without altering the system prompt. Since our primary comparison models (e.g., the Qwen3 family) are all thinking models, we use a consistent prompt template across training and evaluation to ensure a fair and controlled comparison. Only GPT and TigerLLM use the instruct-style prompt template, and those results are presented as secondary comparisons in the paper. The difference between the two prompt variants is minimal—the thinking model prompt

Metric	Value
Bengali characters	88.71%
<i>Non-Bengali token breakdown:</i>	
Math (symbols, expressions, variables)	47.55%
Numerals (incl. currency, percentages)	34.17%
Markdown / formatting	11.20%
Units / compounds (e.g., miles/hours)	5.32%
English words	0.69%
Avg. code-switches per sentence	0.014
English word rate (of all tokens)	0.11%
Top English words	of, gb, time, dvd, tv

Table 6: Code-switching analysis of GANITLLM-4B generations across Bn-MGSM and Bn-MSVAMP (1,236 samples; 101,869 tokens).

	GANIT-DEV \uparrow				MGSM \uparrow	MSVAMP \uparrow	Avg. Accuracy \uparrow	Avg. Words \downarrow	Bn (%) \uparrow
	Easy	Medium	Hard	Olympic					
gpt-4.1	92.83	90.10	84.41	78.18	89.20	82.30	86.17	200	88.16
gpt-4.1-mini	89.69	81.19	83.87	72.73	87.20	78.60	82.21	232	88.18
TigerLLM-9B	45.74	39.11	31.72	31.52	47.20	40.40	39.28	206	93.69
Qwen3-32B	86.55	84.65	79.57	65.45	85.60	76.10	79.65	712	21.08
Qwen3-14B	83.41	81.19	75.81	69.09	83.60	75.80	78.15	767	17.87
Qwen3-8B	77.58	72.40	63.79	56.00	75.12	72.42	69.55	846	16.48
Qwen3-4B	74.89	68.32	60.22	53.33	69.20	70.50	66.08	943	14.79
+ SFT	63.23	50.00	45.70	41.82	74.00	74.60	58.23	184	86.65
+ CGRPO	86.10	77.23	72.04	70.91	82.40	78.50	77.86	844	14.94
+ SFT + GRPO	67.26	50.00	50.00	49.70	77.60	76.30	61.81	189	88.61
+ SFT + CGRPO	69.06	51.49	53.76	47.88	76.80	76.40	62.56	193	88.71
Qwen3-1.7B	25.56	30.69	22.04	15.76	15.20	14.10	20.56	1124	19.64
+ SFT	30.94	28.22	19.89	15.76	48.80	64.60	34.70	253	87.79
+ CGRPO	56.05	53.96	46.24	41.82	59.60	66.20	53.98	1002	18.74
+ SFT + GRPO	33.63	30.69	20.97	24.85	53.60	66.90	38.44	207	88.32
+ SFT + CGRPO	36.32	27.72	19.35	21.82	52.80	66.80	37.47	210	87.80
Qwen3-0.6B	7.17	6.44	6.45	4.85	8.40	12.20	7.59	1265	12.43
+ SFT	11.66	7.92	5.91	12.12	28.40	51.40	19.57	263	88.60
+ CGRPO	13.45	11.39	12.37	9.70	17.20	35.20	16.55	824	11.67
+ SFT + GRPO	14.35	9.41	9.14	8.48	32.40	52.50	21.05	246	88.45
+ SFT + CGRPO	13.90	8.91	10.22	12.73	28.40	52.40	21.09	248	88.70

Table 7: Model Performance on GANIT-DEV, Bn-MGSM (Shi et al., 2023) and Bn-MSVAMP (Chen et al., 2023) test set.

Model	PolyMath-low		PolyMath-medium		MT-MATH100	
	Score	%Bn	Score	%Bn	Score	%Bn
Qwen3-4B	68.80	62.46	52.94	23.95	84.75	38.28
GANITLLM-4B + SFT	77.60	85.53	15.69	74.89	59.32	76.78
GANITLLM-4B + SFT + GRPO	73.60	87.83	17.12	77.18	67.46	80.62
GANITLLM-4B + SFT + CGRPO	80.00	86.59	17.76	77.64	71.19	79.52

Table 8: Results on PolyMath and MT-MATH100 benchmarks.

GANIT	Difficulty Distribution				Source Distribution					
	Split	Easy	Med.	Hard	Oly.	Numina	Somadhan	mCot	BDMO	s1K
GANIT-SFT		10015	84	208	716	7827	3039	157	–	–
GANIT-RLVR		1832	1832	1832	1832	6558	462	271	30	7
GANIT-DEV		223	202	186	165	704	40	27	4	1

Table 9: Difficulty and source distribution of the GANIT dataset. The splits contain 11,023 (GANIT-SFT), 7,328 (GANIT-RLVR), and 776 (GANIT-DEV) samples respectively. BDMO stands for Bangladesh Math Olympiad.

Hyperparameter	Value
Training Configuration	
Training epochs	50
Global batch size	64
Learning rate	1×10^{-6}
Learning rate scheduler	Cosine with Min LR
Minimum learning rate	1×10^{-7}
Warmup ratio	0.05
Gradient clipping norm	1.0
Model Configuration	
Training type	Full fine-tuning
Precision	bfloat16
Max sequence length	4096

Table 10: Supervised Fine-tuning (SFT) Training Hyperparameters

Model ID	Bn-MGSM \uparrow	Bn-MSVAMP \uparrow
Qwen2.5-14B-Instruct	79.60	77.70
Qwen2.5-32B-Instruct	88.80	80.90
Qwen2.5-72B-Instruct	88.40	81.00
Qwen3-8B	88.40	82.10
Qwen3-14B	87.60	83.30
Qwen3-32B	92.00	83.80
gpt-oss-20B	88.40	82.10
Llama-3.3-70B	60.40	77.40

Table 12: Zero-shot evaluation (Pass@4) of MGSM and MSVAMP using 8 recent open-source LLMs.

omits explicit `<think>` tags since these models natively produce structured reasoning traces, while the instruct model prompt includes `<think>` and `<answer>` tags to elicit a similar reasoning format. We believe this slight variation does not significantly affect our core claims.

User Prompt for Training & Evaluation

Thinking Model (Primary: Qwen3 family, our models)

Hyperparameter	Value
Optimization	
Learning Rate	1×10^{-4}
Learning Rate Scheduler	Cosine with Min LR
Minimum Learning Rate	1×10^{-5}
Warmup Ratio	0.05
Training Epochs	5
Global Batch Size	64
Gradient Clipping	1.0
LoRA Configuration	
LoRA Rank	16
LoRA Alpha	32
GRPO Parameters	
Temperature	1.0
Beta (KL Regularization)	0.1
Number of Rollout	8
Max Model Length	4096
Max Completion Length	2500
Loss	DAPO
Dynamic Sample	True
Max Resample Times	3
Epsilon High	0.28
Epsilon Low	0.20

Table 11: Hyperparameters for GRPO LoRA Fine-tuning

A conversation takes place between the user and the assistant. The user asks a question, and the assistant solves the problem. Please reason step by step in Bengali, and put your final answer in the `<answer>` `</answer>` tags.

Question: `{{problem}}`

Instruct Model (Secondary: GPT, TigerLLM)

A conversation takes place between the user and the assistant. The user asks a question, and the assistant solves the problem. Please reason step by step, and put your reasoning process and final answer in the `<think>` `</think>` and `<answer>` `</answer>` tags respectively.

Question: `{{problem}}`

H Use of AI Assistance

The authors have used Cursor⁵ during development and ChatGPT⁶ for proofreading and polishing the final writing. Content provided to those tools were original to the authors.

⁵<https://cursor.com>

⁶<https://chatgpt.com/>

Bengali Problem: তিন বন্ধু টিভির সামনে বসে আছে। জানা গেছে, তাদের প্রত্যেকেই সবসময় সঠিক কথা বলে অথবা সবসময় ভুল কথা বলে। প্রথমজন বললো: "আমাদের কারোই এই ছবিটি দেখা হয়নি।" দ্বিতীয়জন বললো: "আমি এই ছবিটি দেখেছি, কিন্তু তোমরা দুজনই দেখিনি।" তৃতীয়জন বললো: "আমি এই ছবিটি দেখেছি।" যদি জানা থাকে যে তাদের মধ্যে অন্তত একজন সবসময় সঠিক কথা বলে এবং অন্তত একজন সবসময় ভুল কথা বলে, তাহলে তাদের মধ্যে কতজন সবসময় সঠিক কথা বলে?			
English Translation: Three friends are sitting in front of a TV. It is known that each of them either always tells the truth or always lies. The first person said, "None of us has seen this movie." The second person said, "I have seen this movie, but neither of you has." The third person said, "I have seen this movie." If it is known that at least one of them always tells the truth and at least one of them always lies, then how many of them always tell the truth?			
Qwen3-4B	Qwen3-4B + SFT	Qwen3-4B + Curriculum GRPO	GanitLLM-4B + SFT → Curriculum GRPO
<p>Output: Okay, let's try to figure this out. So there are three friends sitting in front of the TV. Each of them either always tells the truth or always lies. The first person says, "None of us have seen this picture." The second says, "I have seen this picture, but you two haven't." The third says, "I have seen this picture." We know that at least one of them is always truthful and at least one is always lying. The question is, how many of them are always truthful?</p> <p>Hmm. Let me break this down. Let's denote the three friends as A, B, and C.</p> <p>First, let's consider the statements:</p> <p>A: "None of us have seen this picture." So A is saying that all three (A, B, C) have not seen the picture.</p> <p>B: "I have seen this picture, but you two haven't." So B is saying that B has seen it, but A and C have not.</p> <p>C: "I have seen this picture." So C is saying that C has seen it.</p> <p>Now, since each person is either a truth-teller (always tells the truth) or a liar (always lies), we need to find a scenario where at least one is truthful and at least one is lying. Also, we need to determine how many are truthful.</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>What if A and C are truthful? Then A's statement is true: none have seen it. C's statement is true: C has seen it. But that's a contradiction because A says none have seen it, but C says he has. Therefore, impossible.</p> <p>What if B and C are truthful? Then B's statement is true: B has seen it, and A and C haven't. But C's statement is true: C has seen it. That's a contradiction because B says C hasn't, but C says he has. So that's impossible.</p> <p>So the only possible scenarios are when exactly one person is truthful and the other two are liars. Therefore, the answer should be 1.</p> <p>Wait, but let me check again. Suppose that two people are truthful. Is there any way that could happen? Let's think again. Suppose that A is lying, B is truthful, and C is lying. Then B's statement is true: B has seen it, and A and C haven't. That's consistent. A's statement is "none have seen it," which is a lie, so at least one has. Which is B. So that works. So in this case, only B is truthful. Similarly, if A is truthful, then B and C are lying. If C is truthful, then A and B are lying. So in all cases, only one is truthful. Therefore, the answer is 1.</p>	<p>Output: 1. **সমস্যার বিবরণ:** - তিনজন বন্ধু আছে যারা প্রত্যেকেই সবসময় সঠিক কথা বলে অথবা সবসময় ভুল কথা বলে। - তারা এই বিবৃতিগুলি দিয়েছে: - প্রথমজন: "আমাদের কারোই এই ছবিটি দেখা হয়নি।" - দ্বিতীয়জন: "আমি এই ছবিটি দেখেছি, কিন্তু তোমরা দুজনই দেখিনি।" - তৃতীয়জন: "আমি এই ছবিটি দেখেছি।" - আমাদের কাছে জানা আছে যে তাদের মধ্যে অন্তত একজন সবসময় সঠিক কথা বলে এবং অন্তত একজন সবসময় ভুল কথা বলে।</p> <p>2. **সম্ভাব্য কেস বিশ্লেষণ:** - আমরা তিনজন বন্ধুকে A, B, এবং C হিসেবে চিহ্নিত করি। - আমাদের কাছে তিনটি সম্ভাব্য কেস আছে: - কেস 1: A সবসময় সঠিক কথা বলে। - কেস 2: A সবসময় ভুল কথা বলে। - কেস 3: A সবসময় সঠিক কথা বলে না এবং সবসময় ভুল কথা বলে না।</p> <p>3. **কেস 1: A সবসময় সঠিক কথা বলে:** - যদি A সবসময় সঠিক কথা বলে, তাহলে তার বিবৃতি সত্য। - অর্থাৎ, "আমাদের কারোই এই ছবিটি দেখা হয়নি।" সত্য। - এর অর্থ হলো B এবং C ছবিটি দেখেনি। - এখন, আমরা B এর বিবৃতি বিশ্লেষণ করি: "আমি এই ছবিটি দেখেছি, কিন্তু তোমরা দুজনই দেখিনি।"</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>- অর্থাৎ, C ছবিটি দেখেছে। - এই ক্ষেত্রে, B এর বিবৃতি "আমি এই ছবিটি দেখেছি, কিন্তু তোমরা দুজনই দেখিনি।" মিথ্যা হবে, যা অসম্ভব কারণ B সবসময় সঠিক কথা বলে না। - অতএব, C সবসময় সঠিক কথা বলে না। - এই কেসে, আমরা দেখতে পাচ্ছি যে শুধুমাত্র A সবসময় সঠিক কথা বলে।</p> <p>5. **কেস 3: A সবসময় সঠিক কথা বলে না এবং সবসময় ভুল কথা বলে না:** - যদি A সবসময় সঠিক কথা বলে না এবং সবসময় ভুল কথা বলে না, তাহলে তার বিবৃতি মিথ্যা বা সত্য হতে পারে। - এই কেসে, আমরা সম্ভাব্য সংমিশ্রণগুলি বিশ্লেষণ করি: - যদি A সবসময় ভুল কথা বলে, তাহলে তার বিবৃতি মিথ্যা। - অর্থাৎ, "আমাদের কারোই এই ছবিটি দেখা হয়নি।" মিথ্যা। - এর অর্থ হলো কমপক্ষে একজন বন্ধু ছবিটি দেখেছে। - এই ক্ষেত্রে, আমরা দেখতে পাচ্ছি যে এই কেসে সমস্যার শর্তগুলি পূরণ হয় না।</p> <p>6. **উপসংহার:** - সমস্ত কেস বিশ্লেষণ করার পরে, আমরা দেখতে পাচ্ছি যে শুধুমাত্র কেস 1 এবং কেস 2 সমস্যার শর্তগুলি পূরণ করে। অতএব, তাদের মধ্যে সবসময় সঠিক কথা বলে এমন সংখ্যা হলো 2</p>	<p>Output: Okay, let's try to figure this out. So there are three friends sitting in front of the TV. Each of them either always tells the truth or always lies. The first person says, "None of us have seen this picture." The second says, "I have seen this picture, but you two haven't." The third says, "I have seen this picture." We know that at least one of them is always truthful and at least one is always lying. The question is, how many of them always tell the truth?</p> <p>Hmm. Let me break this down. Let's denote the three friends as A, B, and C.</p> <p>First, let's consider the statements:</p> <p>A: "None of us have seen this picture." So A is saying that all three (A, B, C) have not seen the picture.</p> <p>B: "I have seen this picture, but you two haven't." So B is saying that B has seen it, but A and C have not.</p> <p>C: "I have seen this picture." So C is saying that C has seen it.</p> <p>Now, since each person is either a truth-teller (always tells the truth) or a liar (always lies), their statements must be consistent with their type.</p> <p>.....</p> <p>.....</p> <p>.....</p> <p>Wait, this is the same contradiction as before. Wait, this is a problem. Because if A is truthful, then none have seen it. But B's lie would require that either B hasn't seen it or C has. But if A is truthful, then C hasn't seen it. Therefore, B's lie would require that B hasn't seen it. So B is lying, and C hasn't seen it. So that's okay. Because B's statement is "I have seen it, but you two haven't." The negation is "Either B hasn't seen it, or A or C has." Since A and C haven't, then B must not have seen it. So B is lying. Therefore, in this case, B is lying, and C is lying. So that's okay. So A is truthful, B and C are liars. So that's possible.</p> <p>But then, in this case, C's statement is "I have seen it." Since C is lying, that's false. So C hasn't seen it. Which is consistent with A's truthful statement. So that's okay.</p> <p>So all three scenarios are possible. Therefore, the answer is that there is exactly one person who is always truthful. But the problem says "at least one is truthful and at least one is lying," so the answer is 1.</p> <p>Wait, but the problem is in Bengali, and the answer is to be given in Bengali. But the user is asking for the answer in the box. So the answer is 1. So the number of people who always tell the truth is 1.</p>	<p>Output: 1. **সমস্যার বিবরণ:** - তিনজন বন্ধু আছে যারা প্রত্যেকেই সবসময় সঠিক কথা বলে অথবা সবসময় ভুল কথা বলে। - তারা এই বিবৃতিগুলি দিয়েছে: - প্রথমজন: "আমাদের কারোই এই ছবিটি দেখা হয়নি।" - দ্বিতীয়জন: "আমি এই ছবিটি দেখেছি, কিন্তু তোমরা দুজনই দেখিনি।" - তৃতীয়জন: "আমি এই ছবিটি দেখেছি।" - আমাদের কাছে জানা আছে যে তাদের মধ্যে অন্তত একজন সবসময় সঠিক কথা বলে এবং অন্তত একজন সবসময় ভুল কথা বলে।</p> <p>2. **সম্ভাব্য ক্ষেত্রগুলি বিশ্লেষণ:** - আমরা তিনজন বন্ধুকে নিম্নলিখিতভাবে চিহ্নিত করি: - A প্রথমজন। - B দ্বিতীয়জন। - C তৃতীয়জন।</p> <p>3. **ক্ষেত্র 1: A সবসময় সঠিক কথা বলে।** - যদি A সবসময় সঠিক কথা বলে, তাহলে তার বিবৃতি সত্য। - অর্থাৎ, তাদের কারোই এই ছবিটি দেখা হয়নি। - এর অর্থ হলো, B এবং C উভয়েই এই ছবিটি দেখেনি। - তবে, B বলেছে, "আমি এই ছবিটি দেখেছি, কিন্তু তোমরা দুজনই দেখিনি।" যদি C সবসময় সঠিক কথা বলে, তাহলে এটি সত্য হবে, কিন্তু এটি বিরোধিতাবাদী কারণ C বলছে যে সে ছবিটি দেখেছে, কিন্তু আমরা ইতিমধ্যেই জানি যে C ছবিটি দেখেনি। - অতএব, B সবসময় ভুল কথা বলে। - C বলেছে, "আমি এই ছবিটি দেখেছি।" যদি C সবসময় সঠিক কথা বলে, তাহলে এটি সত্য হবে, কিন্তু এটি বিরোধিতাবাদী কারণ C বলছে যে সে ছবিটি দেখেছে, কিন্তু আমরা ইতিমধ্যেই জানি যে C ছবিটি দেখেনি। - অতএব, C সবসময় ভুল কথা বলে। - এই ক্ষেত্রে, শুধুমাত্র A সবসময় সঠিক কথা বলে।</p> <p>4. **ক্ষেত্র 2: A সবসময় ভুল কথা বলে।** - যদি A সবসময় ভুল কথা বলে, তাহলে তার বিবৃতি মিথ্যা। - অর্থাৎ, তাদের কারোই এই ছবিটি দেখা হয়েছে। - এর অর্থ হলো, B অথবা C অথবা উভয়েই এই ছবিটি দেখেছে। - ধরুন B সবসময় সঠিক কথা বলে। - B বলেছে, "আমি এই ছবিটি দেখেছি, কিন্তু তোমরা দুজনই দেখিনি।" - যদি B সবসময় সঠিক কথা বলে, তাহলে এটি সত্য হবে, কিন্তু এটি বিরোধিতাবাদী কারণ আমরা ইতিমধ্যেই জানি যে A ছবিটি দেখেছে। - অতএব, B সবসময় ভুল কথা বলে। - ধরুন C সবসময় সঠিক কথা বলে। - C বলেছে, "আমি এই ছবিটি দেখেছি।" - যদি C সবসময় সঠিক কথা বলে, তাহলে এটি সত্য হবে, কিন্তু এটি বিরোধিতাবাদী কারণ আমরা ইতিমধ্যেই জানি যে A ছবিটি দেখেছে। - অতএব, C সবসময় ভুল কথা বলে। - এই ক্ষেত্রে, কেউই সবসময় সঠিক কথা বলে না।</p> <p>5. **উপসংহার:** - যেহেতু আমাদের কাছে জানা আছে যে তাদের মধ্যে অন্তত একজন সবসময় সঠিক কথা বলে, তাই ক্ষেত্র 2 বাদ দেওয়া হয়েছে। - অতএব, শুধুমাত্র ক্ষেত্র 1 বৈধ।</p> <p>অতএব, তাদের মধ্যে ঠিক একজন সবসময় সঠিক কথা বলে।</p> <p>উত্তর: ১</p>
<p>Correct: </p> <p>Number of Words: 932 </p> <p>Bengali Percentage: 7.58% </p>	<p>Correct: </p> <p>Number of Words: 645 </p> <p>Bengali Percentage: 97.63% </p>	<p>Correct: </p> <p>Number of Words: 2223 </p> <p>Bengali Percentage: 7.32% </p>	<p>Correct: </p> <p>Number of Words: 467 </p> <p>Bengali Percentage: 97.7% </p>

Figure 4: Qualitative comparison of training configurations on an Olympiad-level problem from GANIT-DEV.