LANG: Reinforcement Learning for Multilingual Reasoning with Language-Adaptive Hint Guidance

Yuchun Fan[1]*, Bei Li[2]†, Peiguang Li[2], Yilin Wang[1], Yongyu Mu[1], Jian Yang[2], Xin Chen[2], Rongxiang Weng[2], Jingang Wang[2], Xunliang Cai[2], Jingbo Zhu[1,3], Tong Xiao[1,3]†

[1] NLP Lab, School of Computer Science and Engineering, Northeastern University, Shenyang, China

[2]Meituan Inc. [3]NiuTrans Research, Shenyang, China

yuchunfan_neu@outlook.com {xiaotong,zhujingbo}@mail.neu.edu.cn

Abstract

Reinforcement learning has proven effective for enhancing multi-step reasoning in large language models (LLMs), yet its benefits have not fully translated to multilingual contexts. Existing methods struggle with a fundamental trade-off: prioritizing input-language consistency severely hampers reasoning quality, while prioritizing reasoning often leads to unintended language drift toward English. We address this challenge with LANG, a novel framework that leverages language-conditioned hints to guide exploration in non-English reasoning tasks. Our method incorporates two key mechanisms to prevent dependency on these hints: a progressive decay schedule that gradually withdraws scaffolding, and a language-adaptive switch that tailors learning horizons to specific language difficulties. Empirical results on challenging multilingual mathematical benchmarks reveal that LANG substantially enhances reasoning performance without compromising language consistency. Moreover, we show that our framework generalizes beyond mathematics, fostering more consistent language alignment across model layers[1].

1 Introduction

Recent years have witnessed the rise of large reasoning models such as OpenAI-o3 (OpenAI, 2025) and DeepSeek-R1 (Guo et al., 2025), which leverage reinforcement learning with verifiable rewards (RLVR) to incentivize multi-step reasoning capabilities. However, these advances remain largely English-centric, leaving a critical performance gap in non-English settings, especially for low-resource languages (Wang et al., 2025b; Luo et al., 2025).

Beyond accuracy, when it comes to multilingual scenarios, language-consistent reasoning is equally crucial for native-user interpretability. However, as illustrated in Figure 1, current large language models (LLMs) face two coupled challenges. First, when we explicitly require models to reason in the input language, strong reasoning capability in English does not reliably transfer across languages, leading to significant performance degradation in non-English settings. Second, when such language constraints are relaxed, models frequently drift toward English, reducing readability for native users. Consequently, it remains non-trivial to balance language consistency and reasoning performance.

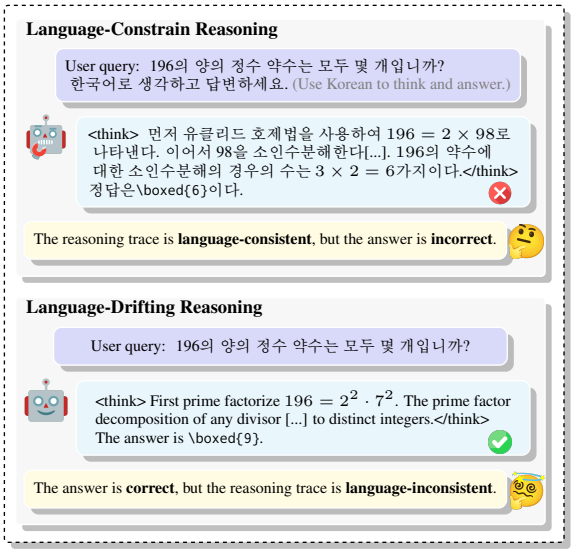

Figure 1: The trade-off between answer accuracy and the language of reasoning trace in multilingual reasoning scenarios. The Korean question in English means "How many positive integer divisors are there of 196?"

To address these challenges, recent studies primarily follow two directions. One line of work introduces explicit prompt constraints to steer models to reason in the language of the user's query during reinforcement learning (RL) rollouts or at inference time (Qi et al., 2025; mGRPO, 2025). However, such prompt-based steering offers limited control and often comes at the cost of reasoning quality.

*Work done during internship at Meituan.

† Corresponding author.

[1]The project will be available at: https://github.com/fmm170/LANG

Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 43838–43866

Building on this, another line of work further introduces language-consistency rewards in RL, encouraging models to align the output language with the input (Zhang et al., 2025d). Nevertheless, when the model's multilingual reasoning ability is limited, trajectories that satisfy such strict objectives are extremely scarce, exacerbating reward sparsity and hindering effective exploration during multilingual reasoning RL.

Inspired by prior studies that leverage hints (*e.g.*, partial reasoning steps from expert demonstrations) to alleviate reward sparsity during RL training (Zhang et al., 2025a; Wang et al., 2025a; Liu et al., 2025), we propose LANG, a language-adaptive hint-guided RL framework for multilingual reasoning. LANG uses language-conditioned hints as scaffolding to bootstrap exploration in non-English settings. However, we find that keeping multilingual hints throughout training induces strong hint dependence, leading to substantial performance degradation when hints are unavailable at test time. To address this, we draw inspiration from scheduled sampling (Bengio et al., 2015; Zhang et al., 2019; Qian et al., 2021), and introduce a progressive hint-decay schedule that reduces hint exposure over training, shifting learning from hint-conditioned rollouts to autonomous multilingual reasoning. Moreover, considering the difference in learning difficulty across languages, we further employ a language-adaptive switch that sets language-specific horizons for turning off hints.

To validate the effectiveness of LANG, we conduct extensive experiments on two representative LLMs across different sizes, covering two challenging multilingual mathematical reasoning benchmarks: MMATH (Luo et al., 2025) and PolyMath (Wang et al., 2025b). The results show that LANG effectively enhances multilingual reasoning ability while preserving language consistency, improving accuracy over vanilla GRPO by 24.1% on MMATH and 18.7% on PolyMath. Moreover, LANG can also be generalized to non-mathematical multilingual tasks. Further analysis highlights that our method achieves consistently language-consistent reasoning across layers.

2 Related Work

2.1 Multilingual Mathematical Reasoning

Recent advances in LRMs have significantly improved their multi-step reasoning capabilities. However, reasoning performance remains highly uneven across languages (Fan et al., 2025a; Zhang et al., 2025b; Zhao and Zhang, 2024), and models often suffer from language drift (*i.e.*, responding in a language different from the user's query), degrading both performance and user experience in non-English settings (Wang et al., 2025b; Luo et al., 2025; Fan et al., 2025b; Wang et al., 2026; Zhang et al., 2025c). Current approaches to enhancing multilingual reasoning under language consistency conditions mainly fall into two categories. One line of work (Qi et al., 2025; Luo et al., 2025) uses explicit constraints in prompts to guide models to reason in the user-desired language. Another line of research (Park et al., 2025; Yang et al., 2025b; mGRPO, 2025; Hwang et al., 2025) introduces language-consistency rewards in reinforcement learning to further encourage alignment between input and output languages. Building upon this, our work attempts to leverage multilingual reasoning traces as explicit guidance to steer model exploration within specific language spaces, a perspective that remains underexplored.

2.2 Hint-guided Reinforcement Learning

A well-known challenge in RLVR is the reward sparsity issue: when task difficulty exceeds model capability, sampled trajectories can all be incorrect, yielding zero group-wise advantage (Yu et al., 2025; Chen et al., 2026; Huang et al., 2026). To mitigate this, prior studies (Zhang et al., 2025a; Wang et al., 2025a; Liu et al., 2025) have focused on injecting hint-like partial solutions into prompts as in-context guidance to improve exploration. This issue is even more pronounced in multilingual reasoning, where non-English reasoning ability typically lags behind English, making it difficult to sample feasible trajectories. Building on this line of work, we introduce a language-adaptive hint decay strategy that scales guidance by language difficulty, effectively breaking the exploration bottleneck in multilingual reasoning.

3 Preliminary Study

In this section, we conduct a pilot study of hint-guided RL for multilingual reasoning and analyze its training dynamics and generation behaviors, providing insights into the design of LANG.

3.1 Experimental Setup for Pilot Study

We adopt QUESTA (Li et al., 2025) as a representative hint-guided approach that augments training prompts with a fixed portion of distilled reasoning

Figure 2: The comparison of Vanilla GRPO and QUESTA during RL training. Blue curves denote Vanilla GRPO and orange curves denote QUESTA. In the middle and right panels, solid lines correspond to the left y-axis, and $*$-marked lines correspond to the right y-axis.

trajectories to guide exploration. We use Qwen2.5-7B-Instruct as the policy model and replace the original trajectories with multilingual counterparts constructed from DeepMath-103K (He et al., 2025). We compare this variant with vanilla GRPO and evaluate its performance on MMATH (Luo et al., 2025). More details are provided in Appendix A.

3.2 Observations

We report two observations below that highlight the limitations of naively applying multilingual hints during RL training.

Finding 1

High policy entropy and training reward do not reliably translate into better test-time multilingual reasoning performance.

As shown in Figure 2 (a–b), QUESTA maintains higher policy entropy and achieves higher reward than vanilla GRPO, yet performs worse at test time. This gap indicates a severe training–inference discrepancy: *the model is optimized under hint-conditioned rollouts during training but must reason autonomously when hints are absent at inference.* The issue is amplified in multilingual scenarios, where reasoning capabilities vary significantly across languages. For low-resource languages, rewarding trajectories are harder to reach without hints, incentivizing reliance on hint-conditioned shortcuts over autonomous multilingual reasoning.

Finding 2

Increased response length does not necessarily indicate stronger reasoning ability and may exacerbate the repeat curse.

Figure 2 (c) shows that QUESTA substantially increases response length while also sharply increasing the repetition score, indicating that the added length is dominated by repetitive generation rather than reflecting richer reasoning patterns (Gandhi et al., 2025). This provides further evidence of the training–inference discrepancy: *once multilingual hints are removed, the model struggles to produce successful trajectories autonomously and degenerates into repetitive patterns*. These results suggest that narrowing the training–inference discrepancy is crucial for transferring multilingual capabilities learned during training to test time.

4 Methodology

In this section, we present LANG, a language-adaptive hint-guided RL framework for multilingual reasoning, as shown in Figure 3. LANG consists of two key components: (1) **Scheduled Multilingual Hint Decay**, which augments training prompts with language-conditioned multilingual reasoning traces and progressively reduces hint exposure to encourage autonomous reasoning; and (2) **Language-adaptive Switch**, which sets language-specific decay horizons to match their varying learning difficulty.

4.1 Scheduled Multilingual Hint Decay

Given a question q in language l, we assume access to a multilingual reasoning trace $h = (h_1, \ldots, h_L)$ produced by a teacher model in the same language, where L is the trace length. At training step t, we inject a prefix of h with length:

$$k_t^l = \left\lfloor p_t^l \, L \right\rfloor, \tag{1}$$

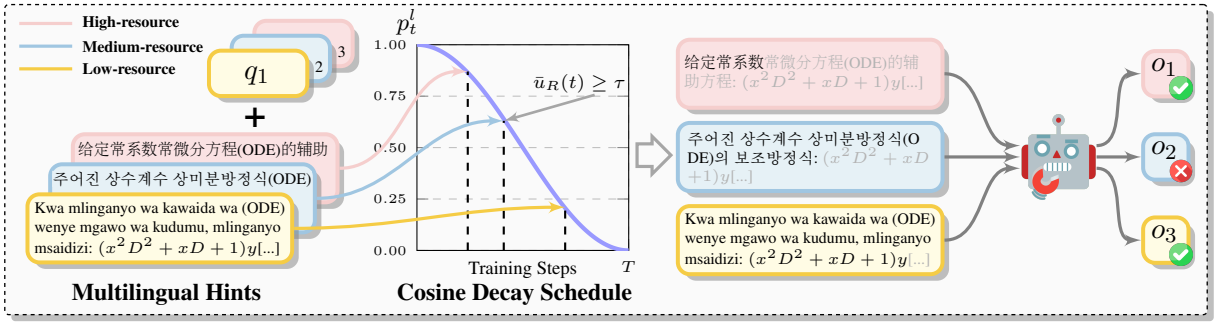

Figure 3: An overview of LANG: our method mitigates reward sparsity by incorporating multilingual hints to guide the model in generating correct multilingual reasoning, combined with a cosine annealing decay schedule and a language-adaptive switch that adjusts hint injection based on each language's learning difficulty.

and construct the hint-conditioned prompt:

$$q_t^l = \begin{cases} q \oplus (h_1, \ldots, h_{k_t^l}), & t \leq T, \\ q, & t > T, \end{cases} \tag{2}$$

where $p_t^l \in [0, 1]$ denotes the hint ratio at step t, and T denotes the step after which hint injection is turned off. By introducing hint guidance to bootstrap exploration, the probability of sampling successful trajectories in non-English settings is effectively increased, thereby mitigating the reward sparsity issue during the early training phase.

Effect of different sampling strategies.

Cosine decay schedule. To avoid hint dependence and narrow the training–inference gap, for $t \leq T$, we instantiate p_t^l using a cosine schedule:

$$p_t^l = \begin{cases} \frac{1}{2}\left(1 + \cos\left(\pi \frac{t}{T}\right)\right), & t \leq T, \\ 0, & t > T. \end{cases} \tag{3}$$

This schedule starts with full guidance ($p_0^l = 1$) and smoothly anneals to zero, yielding a gradual transition from hint-conditioned rollouts to autonomous multilingual reasoning.

4.2 Language-adaptive Switch

Due to the substantial gaps in multilingual reasoning ability across languages, a single global switch step is suboptimal. To this end, we introduce a language-adaptive switching strategy that tailors learning horizons to specific language difficulties. Specifically, we partition languages into resource groups $\mathcal{R} \in \{\text{high}, \text{mid}, \text{low}\}$. For each rollout step t, we compute an effective-update rate $u_{\mathcal{R}}(t)$, defined as the fraction of instances in the group batch $\mathcal{B}_{\mathcal{R}}(t)$ whose rollout group contains at least one trajectory with positive advantage:

$$u_{\mathcal{R}}(t) = \frac{1}{|\mathcal{B}_{\mathcal{R}}(t)|} \sum_{x \in \mathcal{B}_{\mathcal{R}}(t)} \mathbb{I}\Big[\exists\, i \in \{1, \ldots, G\} \text{ s.t. } A_i(x, t) > 0\Big], \tag{4}$$

where $\mathbb{I}$ is the indicator function.

To reduce variance, we adopt an exponential moving average with $\alpha = 0.5$ to control the smoothing strength.

$$\bar{u}_{\mathcal{R}}(t) = \alpha\, \bar{u}_{\mathcal{R}}(t-1) + (1-\alpha)\, u_{\mathcal{R}}(t). \tag{5}$$

A high $\bar{u}_{\mathcal{R}}(t)$ indicates that successful trajectories are consistently reachable, serving as an effective signal to remove scaffolding.

Switch criterion. We switch a resource group $\mathcal{R}$ to the zero-hint regime once $\bar{u}_{\mathcal{R}}(t)$ exceeds a global threshold τ and define the switch step as:

$$T_{\mathcal{R}} = \min\{t \mid \bar{u}_{\mathcal{R}}(t) \geq \tau\}. \tag{6}$$

As a result, low-resource groups retain hints longer to alleviate reward sparsity, while high-resource groups switch earlier to autonomous reasoning, reducing hint dependence.

4.3 Policy Optimization via GRPO

We optimize the policy model π_θ using GRPO (Shao et al., 2024). At each step, we sample a group of G outputs $\{o_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot \mid q_t^l)$ from the current policy, obtain rewards r_i, and compute standardized group-wise advantages:

$$A_i = \frac{r_i - \text{mean}(\{r_1, \ldots, r_G\})}{\text{std}(\{r_1, \ldots, r_G\})}. \tag{7}$$

GRPO then updates π_θ by maximizing the clipped surrogate objective with KL regularization:

$$\mathcal{J}(\theta) = \mathbb{E}\left[\frac{1}{G}\sum_{i=1}^{G}\min\Big(\rho_i A_i,\ \text{clip}(\rho_i, 1-\varepsilon, 1+\varepsilon)A_i\Big)\right], \tag{8}$$

where $\rho_i = \frac{\pi_\theta(o_i|q_t^l)}{\pi_{\theta_{\text{old}}}(o_i|q_t^l)}$. We omit the KL divergence term following the common practice as $\beta = 0$.

Reward. We use a binary, outcome-style reward that jointly enforces (i) *language consistency*, (ii) *output format*, and (iii) *answer correctness*. Given a model output o with a reasoning trace o_t and a final answer o_a, we define: (1) $R_{\text{lc}}(o) = 1$ if both o_t and o_a are in the same language[2] as the input question, and 0 otherwise; (2) $R_{\text{format}}(o) = 1$ if o contains the thinking tag `<think>...</think>` and provides the final answer within `\boxed{}`, and 0 otherwise; (3) $R_{\text{acc}}(o) = 1$ if the extracted final answer matches the ground truth under a rule-based verifier, and 0 otherwise. The overall reward is the conjunction:

$$R(o) = \mathbb{I}\big[R_{\text{lc}}(o) = 1 \wedge R_{\text{format}}(o) = 1 \wedge R_{\text{acc}}(o) = 1\big]. \tag{9}$$

5 Experimental Setups

5.1 Evaluation Datasets

To assess the effectiveness of LANG, our main experiments are conducted on two challenging multilingual mathematical reasoning benchmarks. Detailed dataset statistics are provided Appendix B.1

MMATH (Luo et al., 2025) comprises problems in ten languages, translated from the AIME[3], CNMO[4], and MATH-500 (Lightman et al., 2024) using GPT-4o-mini (OpenAI, 2023).

PolyMath (Wang et al., 2025b) spans eighteen languages and four easy-to-hard difficulty levels, providing a comprehensive assessment of multilingual reasoning capabilities.

5.2 Evaluation Metrics

Following Wang et al. (2025b), our evaluation primarily focuses on : *language consistency*, *accuracy*, and their conjunction. Detailed definitions are provided in Appendix B.2.

Language Consistency Ratio (LCR). LCR measures whether the model responds in the same language as the input question, computed based on the language-consistency criterion in Equation 9.

Accuracy (Acc). We compute accuracy by comparing the extracted final answer with ground-truth, regardless of response language (Luo et al., 2025).

Language Consistency & Accuracy (LC&Acc). LC&Acc is our primary metric. It counts an output as correct only if the final answer is correct and the entire response is language-consistent with the input (Wang et al., 2025b).

5.3 Baselines

We compare LANG against two categories of baselines: *prompting-based* and *training-based*. See Appendix H for implementation details.

5.3.1 Prompting-based Methods

Language-Constraint Prompting (LCP) (Wang et al., 2025b), **Discourse-Initiated Thinking (DIT)** (Luo et al., 2025), and **Question-Restatement Thinking (QRT)** (Luo et al., 2025) simply prompt LLMs to generate language-constraint reasoning responses. Detailed implementation of these methods refers to Appendix H.1.

5.3.2 Training-based Methods

Multilingual Supervised Fine-tuning (M-SFT) fine-tunes models on constructed multilingual mathematical data to improve performance.

Vanilla GRPO (Guo et al., 2025) applies the standard GRPO algorithm with format and accuracy rewards to perform RL training on multilingual mathematical data.

Language-Consistency GRPO (LC-GRPO) extends GRPO algorithm with a language-consistency reward, constraining the model to respond in the language of questions.

M-Thinker (Zhang et al., 2025d) extends GRPO algorithm by evaluating cross-lingual thinking alignment with an LLM-as-judge and adding a language-consistency reward, thereby enhancing the multilingual reasoning alignment.

mGRPO (mGRPO, 2025) encourages the model to sample multilingual trajectories within each group via prompting, mitigating drift toward English in the reasoning traces.

[2] Following Wang et al. (2025b), we use the `langdetect` library to identify the language.

[3] https://maa.org/maa-invitational-competitions/

[4] https://www.cms.org.cn/

Method	In-Domain Languages							Out-of-Domain Languages					ALL-Avg.
	Ar	Th	Fr	Ja	Zh	En	Avg.	Vi	Ko	Pt	Es	Avg.	
Qwen2.5-3B-Instruct	3.0	0.9	0.7	2.1	13.7	23.0	7.2	5.8	0.9	0.4	5.1	3.1	5.6
	13.1	11.0	16.7	13.7	15.1	23.0	15.4	15.9	13.0	18.5	16.6	16.0	15.7
+ LCP	13.7	16.6	16.4	12.4	14.3	21.3	15.8	13.1	10.9	15.8	13.3	13.3	14.8
+ DIT	12.5	17.6	17.0	12.3	15.7	**23.2**	16.4	14.9	12.3	17.7	12.4	14.3	15.6
+ QRT	12.4	18.5	17.2	13.2	16.6	22.3	16.7	14.0	12.2	16.7	10.9	13.5	15.4
+ M-SFT	12.8	18.9	18.5	13.1	15.3	22.4	16.8	13.3	11.9	16.4	12.1	13.4	15.5
+ Vanilla GRPO	2.5	4.2	9.0	6.9	**17.2**	22.4	10.4	4.6	1.1	1.1	1.3	9.5	7.3
+ LC-GRPO	14.3	17.7	19.4	14.6	14.9	22.9	17.3	17.3	13.7	**18.2**	15.1	16.1	16.8
+ M-Thinker	12.8	17.5	19.6	11.8	13.8	22.4	16.3	14.6	12.1	17.1	11.1	13.7	15.3
+ mGRPO	**15.0**	15.3	18.2	12.2	14.1	19.9	15.8	16.3	14.5	14.9	**16.7**	15.6	15.7
+ LANG	14.5	**19.5**	**20.7**	**15.4**	16.9	23.1	**18.3**	**19.6**	**14.8**	18.1	14.6	**16.8**	**17.7**
Qwen2.5-7B-Instruct	0.3	0.5	0.2	3.6	21.0	28.2	9.0	1.3	1.8	0.7	2.1	1.5	6.0
	20.3	21.5	24.1	25.4	23.8	28.2	23.9	20.8	24.6	24.1	25.1	23.6	23.8
+ LCP	18.0	16.1	23.2	18.6	19.5	29.6	20.8	19.8	19.8	23.4	22.9	21.5	21.1
+ DIT	18.3	16.8	19.2	19.8	19.4	29.0	20.4	18.4	13.3	19.8	21.4	18.2	19.5
+ QRT	17.7	16.1	20.3	19.8	23.1	27.7	20.8	22.4	14.3	16.2	23.6	19.1	20.1
+ M-SFT	12.9	15.2	12.6	19.8	19.4	32.4	18.7	12.0	16.3	16.3	17.1	15.4	17.4
+ Vanilla GRPO	0.0	0.7	0.0	0.1	0.7	30.9	5.4	0.2	0.0	0.0	0.0	0.1	3.3
+ LC-GRPO	24.3	23.6	28.2	25.0	24.7	30.2	26.0	25.2	24.9	29.2	27.8	26.8	26.3
+ M-Thinker	21.4	20.7	22.7	24.5	**25.7**	28.6	23.9	24.6	22.8	26.7	26.7	25.2	24.4
+ mGRPO	24.7	20.5	27.8	26.5	20.1	31.0	25.1	25.9	**25.0**	29.9	28.6	27.3	26.0
+ LANG	**26.3**	**28.5**	**31.1**	**28.0**	22.5	**32.1**	**28.1**	**30.1**	24.5	**32.2**	**31.2**	**29.5**	**28.6**

Table 1: The LC&Acc (%) on MMATH test sets. "Avg." denotes the average performance within each split, and "ALL-Avg." denotes the overall average across all languages. The highest score among systems of the same size is highlighted in **bold**. And gray-colored text indicates the accuracy (%) without considering language consistency.

5.4 Experimental Details

We conduct experiments on the multilingual DeepMath-103K dataset, selecting ten languages in PolyMath as in-domain and reserving the remaining languages for out-of-domain evaluation. Subsequently, for each in-domain language, we sample 0.3K instances for cold-start training and then sample an additional 3K instances for RL training from the remaining data. To verify the effectiveness of our method, we conduct experiments on Qwen2.5-3B/7B/32B-Instruct (Yang et al., 2025a) and Llama3.1-8B-Instruct (AI@Meta, 2024). For more details about training, refer to Appendix C.

6 Experimental Results

6.1 Main Results

We present the main metric LC&Acc on the MMATH and PolyMath benchmarks in Table 1 and 2, respectively. Detailed results for LCR and Accuracy are provided in Appendix D. The results of Qwen2.5-32B-Instruct and Llama3.1-8B-Instruct refer to Appendix F.

Current LLMs struggle to generate language-consistent reasoning traces while preserving accuracy. As shown in Tables 1 and 2, Qwen-family models achieve strong English accuracy, yet their LC&Acc degrades substantially once the intermediate reasoning traces are required to remain in the input language. Moreover, existing approaches further reveal a consistent trade-off between language consistency and accuracy. In particular, on Qwen2.5-3B-Instruct, prompting-based methods can greatly increase language consistency (e.g., QRT boosts the MMATH LCR from 23.3% to 85.1%), but this gain comes with an 11.3% drop in accuracy. Conversely, training-based approaches without explicit language constraints achieve notable accuracy improvements but reduce language consistency. Specifically, on PolyMath with Qwen2.5-7B-Instruct, vanilla GRPO yields a 9.0% relative accuracy gain, while its LCR drops by 16.5% relative to QRT. Overall, these results indicate that generating language-consistent reasoning traces with correct answers remains challenging.

LANG effectively improves multilingual reasoning ability without sacrificing language consistency. As shown in Tables 1 and 2, LANG consistently outperforms all competitive baselines on both MMATH and PolyMath in the in-domain setting. Across the four evaluation models, LANG improves average LC&Acc by 24.1% on MMATH and 18.7% on PolyMath relative to LC-GRPO. Moreover, LANG achieves near-perfect language consistency as illustrated in Appendix D (Figure 11-14). Remarkably, these gains are particularly pro-

Method	In-Domain Languages											Out-of-Domain Languages								ALL-Avg.
	Ar	Bn	Th	Sw	Ja	Zh	De	Fr	Ru	En	Avg.	Te	Ko	Vi	It	Id	Pt	Es	Avg.	
Qwen2.5-3B-Instruct	8.3	4.5	7.5	1.3	8.5	7.9	9.2	9.6	8.7	11.4	7.7	1.0	8.4	9.9	9.5	9.6	11.4	11.2	8.7	8.1
	8.4	4.5	7.6	1.3	8.6	9.4	9.3	9.6	8.9	11.4	7.9	1.0	8.4	9.9	10.0	9.7	11.8	11.4	8.9	8.3
+ LCP	8.2	3.6	8.4	**3.4**	6.8	10.2	9.9	9.2	8.8	**14.1**	8.3	2.4	7.4	9.2	9.5	9.8	10.9	9.9	8.4	8.3
+ DIT	7.6	5.3	6.4	3.1	7.4	9.2	**11.7**	9.9	10.3	11.7	8.3	1.0	8.2	8.8	9.5	9.2	10.2	10.2	8.2	8.2
+ QRT	**10.8**	3.2	6.3	1.5	9.2	6.9	9.5	10.9	10.8	12.7	8.2	1.5	7.5	11.4	10.6	10.5	9.4	9.7	8.7	8.4
+ M-SFT	2.7	2.8	4.3	0.9	2.7	4.6	3.0	2.9	5.9	4.5	3.4	0.4	5.0	5.4	3.1	2.2	3.7	5.7	3.6	3.5
+ Vanilla GRPO	9.1	0.4	8.1	1.5	8.5	10.7	10.2	11.1	10.6	12.7	8.3	2.6	8.5	9.7	7.8	**12.2**	10.0	**11.6**	8.9	8.6
+ LC-GRPO	9.9	6.6	10.5	3.0	9.7	10.8	9.9	11.7	10.2	12.4	9.5	3.0	8.5	10.5	10.8	11.9	11.5	10.9	9.6	9.5
+ M-Thinker	9.1	4.8	6.5	2.3	7.3	8.7	9.7	8.3	10.7	12.5	8.0	1.1	8.1	8.9	9.4	7.5	9.8	9.2	7.7	7.9
+ mGRPO	8.7	5.8	8.7	1.0	7.5	8.3	10.0	**11.7**	9.0	12.7	8.4	3.1	7.8	8.6	10.1	8.0	8.4	10.1	8.0	8.2
+ LANG	8.9	**7.4**	**10.8**	2.7	**9.9**	**13.5**	10.9	11.1	**11.6**	13.3	**10.0**	**4.5**	**9.6**	**12.2**	**11.8**	10.5	**12.1**	10.9	**10.2**	**10.1**
Qwen2.5-7B-Instruct	12.1	11.8	12.7	2.2	13.8	15.3	15.3	15.2	14.8	17.3	13.1	6.2	12.4	14.4	13.4	9.7	13.6	15.1	12.0	12.7
	12.5	12.0	13.3	5.0	13.8	17.4	16.2	15.4	15.7	17.3	13.9	6.2	12.5	14.5	13.4	13.9	14.8	15.7	13.0	13.5
+ LCP	9.0	5.1	8.3	1.3	8.2	6.8	8.9	9.3	10.9	14.2	8.2	1.0	9.4	9.2	10.0	7.9	10.5	10.7	8.4	8.3
+ DIT	12.1	10.8	11.8	2.8	13.2	13.7	13.5	16.6	15.7	16.6	12.7	6.6	11.6	11.9	16.8	13.5	14.4	14.3	12.7	12.7
+ QRT	12.2	**11.5**	12.1	**4.9**	14.5	13.4	14.8	16.1	15.0	17.5	13.2	7.7	11.5	14.0	13.6	13.7	14.9	16.5	13.1	13.2
+ M-SFT	5.1	2.8	6.7	1.3	4.8	3.7	4.1	5.1	5.3	6.1	4.5	1.4	4.4	6.5	7.1	3.4	7.5	5.9	5.2	4.8
+ Vanilla GRPO	13.1	4.5	1.4	2.1	13.1	**15.8**	16.6	15.9	15.3	17.4	11.5	4.2	12.4	15.8	15.6	14.1	16.4	16.2	13.5	12.3
+ LC-GRPO	14.3	9.0	11.2	2.5	13.7	13.4	17.2	17.0	18.3	17.6	13.4	6.0	13.3	16.5	16.6	14.5	17.1	18.7	14.7	13.9
+ M-Thinker	13.6	8.8	9.0	3.1	11.7	12.1	13.2	14.8	14.7	14.3	11.5	**7.8**	10.8	12.3	14.3	13.4	13.8	15.5	12.6	12.0
+ mGRPO	14.9	10.6	11.5	2.7	12.7	13.1	13.8	15.4	16.4	15.3	12.6	6.5	12.8	13.8	15.7	13.3	16.4	16.5	13.6	13.0
+ LANG	**16.2**	11.0	**15.3**	3.4	**15.3**	15.7	**17.9**	**17.9**	**19.4**	**17.9**	**15.0**	7.6	**16.1**	**17.2**	**20.5**	**17.4**	**17.7**	**19.0**	**16.5**	**15.6**

Table 2: The LC&Acc (%) on PolyMath test sets.

nounced in low-resource languages. For example, with Qwen2.5-7B-Instruct, LANG yields LC&Acc gains of 39.0% on Thai in MMATH and 24.6% on Vietnamese in PolyMath over mGRPO. This indicates that LANG leverages fine-grained multilingual hints with explicit language-consistency supervision, mitigating the trade-off between accuracy and language consistency.

LANG exhibits strong out-of-distribution and cross-model generalization. LANG consistently improves LC&Acc across diverse evaluation languages and outperforms all strong baselines on out-of-domain languages as shown in Tables 1 and 2. Specifically, LANG achieves average LC&Acc gains of 79.3% on Llama3.1-8B-Instruct and 11.2% on Qwen2.5-7B-Instruct over LC-GRPO across two benchmarks. This superior generalization demonstrates that the multilingual reasoning ability learned by LANG can be effectively transferred to unseen languages. Additionally, we note consistent improvements across model families and scales. As shown in Tables 8 and 9, LANG achieves average LC&Acc gains over LC-GRPO of 41.1% for Llama3.1-8B-Instruct and 4.9% for Qwen2.5-32B-Instruct across two benchmarks. These consistent gains across languages and model variants demonstrate the robustness of LANG in guiding models to sample language-consistent reasoning trajectories.

6.2 Ablation Study

We conduct ablation studies to validate LANG, with additional ablations reported in Appendix E.

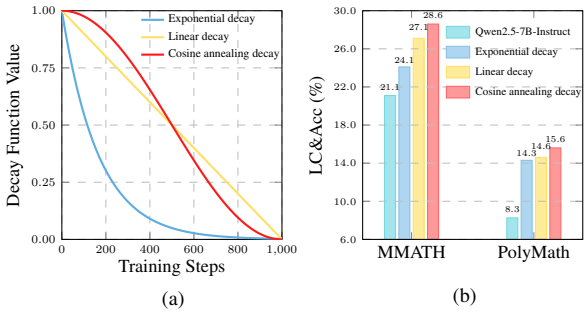

Figure 4: (a) Examples of decay schedules (b) The average performance on MMATH and PolyMath dataset.

Effect of different sampling strategies. To evaluate the effectiveness of the cosine annealing sampling strategy, we conduct ablation studies on Qwen2.5-7B-Instruct by comparing it with exponential and linear decay. As shown in Figure 4 (a), exponential decay reduces the hint length too quickly in the early stage, whereas linear decay applies a uniform schedule that ignores the model's evolving reasoning abilities. In contrast, cosine annealing better matches the desired pattern by providing stronger guidance early and encouraging more autonomous exploration later, thereby achieving the best average LC&Acc on both benchmarks in Figure 4 (b). These results suggest that cosine annealing alleviates early reward sparsity while enabling a timely transition to independent exploration, thus improving consistency between training and inference.

Effect of key ingredients in the LANG training framework. To further assess the necessity

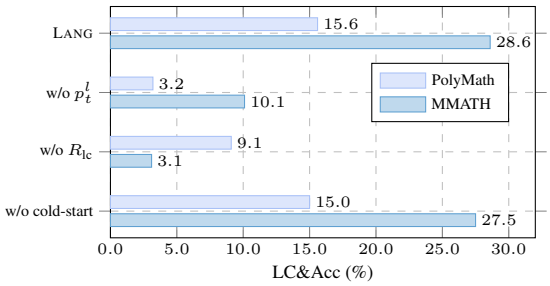

Figure 5: The average performance of training different components on MMATH and PolyMath datasets.

Method	MMLU-ProX	XWinograd	XStoryCloze	XCOPA
Qwen2.5-7B-Instruct	35.9	65.7	60.9	60.7
+ Vanilla GRPO	21.0	65.7	57.5	58.9
+ LC-GRPO	39.9	62.2	60.2	60.5
+ LANG	41.0↑**14.2%**	79.9↑**21.6%**	63.5↑**4.3%**	62.9↑**3.6%**

Table 3: The average extract match and accuracy (%) on non-mathematical benchmarks. **Arrows** indicate the relative improvement over Qwen2.5-7B-Instruct model.

of the language-consistency reward, the hint annealing phase, and cold-start training, we conduct ablation studies on Qwen2.5-7B-Instruct. As shown in Figure 5, without the hint annealing phase p_t^l, maintaining full hints throughout training amplifies the training inference discrepancy and consequently degrades performance. Furthermore, removing the language-consistency reward R_{lc} causes pronounced language drifting of the reasoning traces into English, undermining the goal of language-consistency reasoning. Additionally, removing the cold-start stage results in a 3.0% average performance drop, indicating that cold-start training improves the initial policy's compliance with instruction-specified output language and formatting constraints, thereby stabilizing and facilitating subsequent multilingual RL training.

7 Further Analysis

7.1 Scalability of LANG Beyond Multilingual Mathematical Reasoning

To evaluate the scalability of LANG, we extend our method on MMLU-ProX (Xuan et al., 2025), XWinograd (Muennighoff et al., 2022; Tikhonov and Ryabinin, 2021), XStoryCloze (Lin et al., 2021), XCOPA (Ponti et al., 2020) benchmarks, which cover multilingual understanding and generation tasks across domains. We utilize lm-evaluation-harness[5] as our evaluation framework

[5] https://github.com/EleutherAI/lm-evaluation-harness

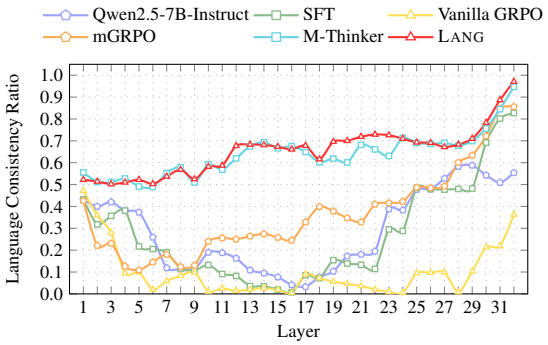

Figure 6: The layer-wise comparison of the language-consistency rate of intermediate decoded outputs with the question language across different training methods.

Translation Model	MMATH	PolyMath
GPT-4o-mini (OpenAI, 2023)	28.6	15.6
Claude-Opus-4.5 (Anthropic, 2025)	27.9	15.0
NLLB-200-3.3B (Costa-jussà et al., 2022)	25.1	13.8

Table 4: The average LC&Acc (%) on MMATH and PolyMath under different translation models for multilingual training data construction, using Qwen2.5-7B-Instruct as the backbone.

(Detail results are provided in Appendix G). Notably, as shown in Table 3, LANG achieves significant average performance improvements of 10.9% over four benchmarks. The results demonstrate that LANG transfers effectively to diverse multilingual tasks, suggesting that multilingual guidance during RL training improves general multilingual capability beyond the target reasoning domain.

7.2 Impact of Translation Quality on Multilingual Performance

We further extend our method by constructing the multilingual training data with different translation models to examine the impact of translation quality on the resulting gains in multilingual performance. As shown in Table 4, the model's multilingual gains remain stable across training data constructed with different translation models. This indicates that LANG does not rely heavily on translation quality, but instead helps the model internalize multilingual reasoning patterns, leading to strong robustness.

7.3 Layer-wise Analysis of Language-Consistent Reasoning

To examine the extent to which the model maintains language-consistent reasoning across layers, rather than reasoning in English in intermediate layers and translating into the target language only at upper layers, which leads to inconsistency in mul-

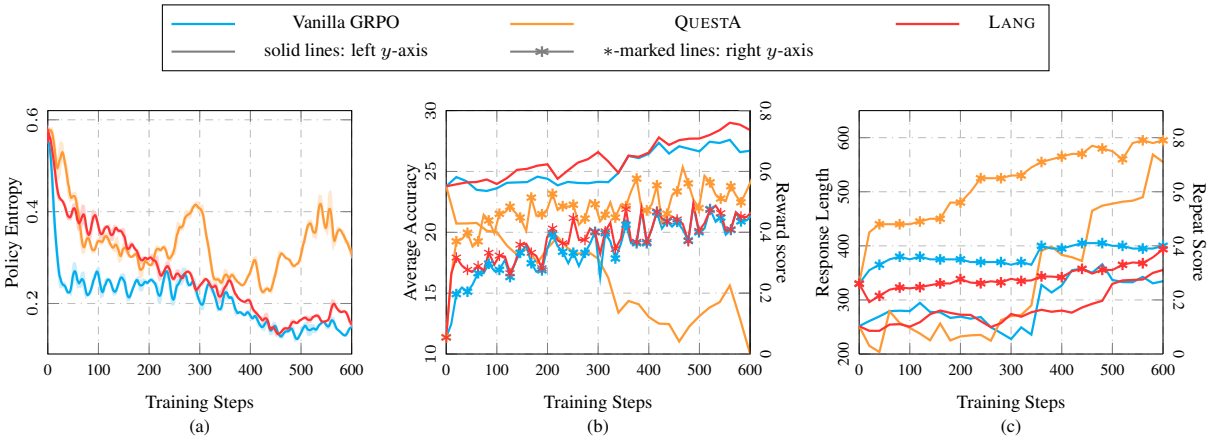

Figure 7: The comparison of Vanilla GRPO and QUESTA during RL training. Blue curves denote Vanilla GRPO, orange curves denote QUESTA, and red curves denote LANG. In the middle and right panels, solid lines correspond to the left y-axis, and $*$-marked lines correspond to the right y-axis.

tilingual reasoning and hinders the model's ability to maintain coherent and accurate reasoning across languages (Zhao et al., 2024). We decode intermediate hidden states into layer-wise outputs with the logit lens (Nostalgebraist, 2020) and measure language consistency with the input question on MMATH. As shown in Figure 6, LANG maintains consistently high language-consistency rates across layers, thereby mitigating disrupted reasoning continuity induced by cross-layer language switching. This result confirms that our method encourages the model to internalize language-consistent reasoning throughout its intermediate representations, rather than relying on superficial language conversion at the final output layer.

7.4 Revisiting Findings after Training with LANG

To facilitate comparisons, we compare the metrics discussed in Section 3.2 before and after training with LANG. As shown in Figure 7, our method attains a higher average reward score during RL training compared to Vanilla GRPO. Furthermore, the higher reward reliably translated into better test-time multilingual reasoning performance. Additionally, LANG increases response length without inducing repetitive generation. This highlights that LANG narrows the training–inference discrepancy, enabling the multilingual capabilities learned during training to transfer effectively to test time.

7.5 Impact of Teacher Model on Multilingual Performance

To verify that our method is not dependent on a specific teacher model, we further extend our experiments by replacing DeepSeek-R1 with GPT-4o-mini to generate reasoning-trace hints for constructing the multilingual training data. As shown in Table 6 and Table 7, the model's multilingual gains remain stable across training data constructed with different teacher models. Specifically, after switching to GPT-4o-mini, LANG still brings improvements of +22.5 and +2.8 points on MMATH and PolyMath respectively, compared to the Qwen2.5-7B-Instruct baseline. This indicates that LANG does not rely heavily on the choice of teacher reasoning model, but instead helps the student model internalize multilingual reasoning patterns from diverse reasoning traces, confirming that our approach is *teacher-agnostic*.

8 Conclusion

In this work, we propose LANG, a language-adaptive hint-guided RL framework for multilingual reasoning. LANG bootstraps exploration with language-conditioned multilingual hints in early training, then progressively reduces hint exposure to mitigate the training-inference discrepancy. A key innovation is our language-adaptive switch, which adjusts the learning pace based on difficulty, specifically preserving guidance for languages that need it while accelerating independence for those that do not. Experiments on MMATH and PolyMath across two LLM families and multiple scales show that LANG substantially improves multilingual reasoning while preserving language consistency. Ablations further verify the necessity of both progressive hint decay and the language-adaptive switching mechanism.

Limitations

Our work presents several limitations worth noting. First, for constructing multilingual hints, we use the widely adopted DeepSeek-R1 as the sole distillation source. Future work will involve extending our method by using data distilled from additional teacher models to more comprehensively evaluate the generalizability of LANG. Second, while our method achieves substantial improvements in accuracy while maintaining language consistency between input and output across both in-domain and out-of-domain test sets across model families and scales, the degrees of performance across different languages result in performance trade-offs. We hypothesize that these trade-offs may arise from imbalances in multilingual training data during the pre-training stage. However, our work starts from the perspective of RL training to enhance the multilingual ablities of LLMs. In future work, we will explore data-centric strategies for improving multilingual capabilities, with a particular focus on data selection and data augmentation.

Ethics Statement

This work does not require ethical considerations. All the data used in this paper is sourced from open-source materials. Throughout the experimental process, all data and models were strictly utilized following their intended purposes and respective licenses. Additionally, this paper may contain offensive text related to the case study. We have all referenced them elliptically and will not present the complete harmful content within the paper.

Acknowledgements

In particular, we sincerely thank Lei Huang for his insightful and constructive suggestions. His generous encouragement has been instrumental in sustaining my efforts to complete this work. This work was supported in part by the National Natural Science Foundation of China (Nos. U24A20334 and 62276056), the Yunnan Fundamental Research Projects (No.202401BC070021), the Yunnan Science and Technology Major Project (No. 202502AD080014), the Fundamental Research Funds for the Central Universities (Nos. N25BSS054 and N25BSS094), and the Program of Introducing Talents of Discipline to Universities, Plan 111 (No.B16009).

References

AI@Meta. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Anthropic. 2025. Claude opus 4.5. https://www.anthropic.com/news/claude-opus-4-5. Accessed: 2025-12-30.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. *Advances in neural information processing systems*, 28.

Hao Chen, Ye He, Yuchun Fan, Yukun Yan, Zhenghao Liu, Qingfu Zhu, Maosong Sun, and Wanxiang Che. 2026. Know more, know clearer: A meta-cognitive framework for knowledge augmentation in large language models. *arXiv preprint arXiv:2602.12996*.

Hao Chen, Yukun Yan, Sen Mei, Wanxiang Che, Zhenghao Liu, Qi Shi, Xinze Li, Yuchun Fan, Pengcheng Huang, Qiushi Xiong, Zhiyuan Liu, and Maosong Sun. 2025. ClueAnchor: Clue-anchored knowledge reasoning exploration and optimization for retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 19258–19278, Suzhou, China. Association for Computational Linguistics.

Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Y. Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loïc Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, John Hoffman, Semarley Jarrett, Kaushik Ram Sadagopan, Dirk Rowe, Shannon Spruit, Chau Tran, Pierre Andrews, Necip Fazil Ayan, Shruti Bhosale, Sergey Edunov, Angela Fan, Cynthia Gao, Vedanuj Goswami, Francisco Guzmán, Philipp Koehn, Alexandre Mourachko, Christophe Ropers, Safiyyah Saleem, Holger Schwenk, and Jeff Wang. 2022. No language left behind: Scaling human-centered machine translation. *CoRR*, abs/2207.04672.

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

Yuchun Fan, Yongyu Mu, YiLin Wang, Lei Huang, Junhao Ruan, Bei Li, Tong Xiao, Shujian Huang, Xiaocheng Feng, and Jingbo Zhu. 2025a. SLAM: Towards efficient multilingual reasoning via selective language alignment. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 9499–9515, Abu Dhabi, UAE. Association for Computational Linguistics.

Yuchun Fan, Yilin Wang, Yongyu Mu, Lei Huang, Bei Li, Xiaocheng Feng, Tong Xiao, and JingBo Zhu. 2025b. Language-specific layer matters: Efficient multilingual enhancement for large vision-language

models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 12473–12500, Suzhou, China. Association for Computational Linguistics.

Kanishk Gandhi, Ayush Chakravarthy, Anikait Singh, Nathan Lile, and Noah D. Goodman. 2025. Cognitive behaviors that enable self-improving reasoners, or, four habits of highly effective stars. *CoRR*, abs/2503.01307.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Zhiwei He, Tian Liang, Jiahao Xu, Qiuzhi Liu, Xingyu Chen, Yue Wang, Linfeng Song, Dian Yu, Zhenwen Liang, Wenxuan Wang, et al. 2025. Deepmath-103k: A large-scale, challenging, decontaminated, and verifiable mathematical dataset for advancing reasoning. *arXiv preprint arXiv:2504.11456*.

Lei Huang, Xiang Cheng, Chenxiao Zhao, Guobin Shen, Junjie Yang, Xiaocheng Feng, Yuxuan Gu, Xing Yu, and Bing Qin. 2026. Bootstrapping exploration with group-level natural language feedback in reinforcement learning. *arXiv preprint arXiv:2603.04597*.

Lei Huang, Xiaocheng Feng, Weitao Ma, Yuchun Fan, Xiachong Feng, Yuxuan Gu, Yangfan Ye, Liang Zhao, Weihong Zhong, Baoxin Wang, Dayong Wu, Guoping Hu, Lingpeng Kong, Tong Xiao, Ting Liu, and Bing Qin. 2025a. Alleviating hallucinations from knowledge misalignment in large language models via selective abstention learning. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 24564–24579. Association for Computational Linguistics.

Lei Huang, Xiaocheng Feng, Weitao Ma, Yuchun Fan, Xiachong Feng, Yangfan Ye, Weihong Zhong, Yuxuan Gu, Baoxin Wang, Dayong Wu, Guoping Hu, and Bing Qin. 2025b. Improving contextual faithfulness of large language models via retrieval heads-induced optimization. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2025, Vienna, Austria, July 27 - August 1, 2025*, pages 16896–16913. Association for Computational Linguistics.

Lei Huang, Xiaocheng Feng, Weitao Ma, Yuxuan Gu, Weihong Zhong, Xiachong Feng, Weijiang Yu, Weihua Peng, Duyu Tang, Dandan Tu, and Bing Qin. 2024a. Learning fine-grained grounded citations for attributed large language models. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 14095–14113. Association for Computational Linguistics.

Lei Huang, Xiaocheng Feng, Weitao Ma, Liang Zhao, Yuchun Fan, Weihong Zhong, Dongliang Xu, Qing Yang, Hongtao Liu, and Bing Qin. 2024b. Advancing large language model attribution through self-improving. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 3822–3836. Association for Computational Linguistics.

Lei Huang, Weijiang Yu, Weitao Ma, Weihong Zhong, Zhangyin Feng, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2025c. A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions. *ACM Trans. Inf. Syst.*, 43(2):42:1–42:55.

Jaedong Hwang, Kumar Tanmay, Seok-Jin Lee, Ayush Agrawal, Hamid Palangi, Kumar Ayush, Ila Fiete, and Paul Pu Liang. 2025. Learn globally, speak locally: Bridging the gaps in multilingual reasoning. *arXiv preprint arXiv:2507.05418*.

Pratik Joshi, Sebastin Santy, Amar Budhiraja, Kalika Bali, and Monojit Choudhury. 2020. The state and fate of linguistic diversity and inclusion in the NLP world. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6282–6293, Online. Association for Computational Linguistics.

Hector J. Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Principles of Knowledge Representation and Reasoning: Proceedings of the Thirteenth International Conference, KR 2012, Rome, Italy, June 10-14, 2012*. AAAI Press.

Jiazheng Li, Hongzhou Lin, Hong Lu, Kaiyue Wen, Zaiwen Yang, Jiaxuan Gao, Yi Wu, and Jingzhao Zhang. 2025. Questa: Expanding reasoning capacity in llms via question augmentation. *arXiv preprint arXiv:2507.13266*.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona Diab, Veselin Stoyanov, and Xian Li. 2022. Few-shot learning with multilingual generative language models. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 9019–9052, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Xi Victoria Lin, Todor Mihaylov, Mikel Artetxe, Tianlu Wang, Shuohui Chen, Daniel Simig, Myle Ott, Naman Goyal, Shruti Bhosale, Jingfei Du, Ramakanth Pasunuru, Sam Shleifer, Punit Singh Koura, Vishrav Chaudhary, Brian O'Horo, Jeff Wang, Luke Zettlemoyer, Zornitsa Kozareva, Mona T. Diab, Veselin Stoyanov, and Xian Li. 2021. Few-shot learning with multilingual language models. *CoRR*, abs/2112.10668.

Mingyang Liu, Gabriele Farina, and Asuman Ozdaglar. 2025. Uft: Unifying supervised and reinforcement fine-tuning. *arXiv preprint arXiv:2505.16984*.

Wenyang Luo, Wayne Xin Zhao, Jing Sha, Shijin Wang, and Ji-Rong Wen. 2025. MMATH: A multilingual benchmark for mathematical reasoning. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 11187–11202, Suzhou, China. Association for Computational Linguistics.

mGRPO. 2025. mGRPO: Unlocking llm reasoning through multilingual thinking. OpenReview. OpenReview submission.

Swaroop Mishra, Arindam Mitra, Neeraj Varshney, Bhavdeep Singh Sachdeva, Peter Clark, Chitta Baral, and Ashwin Kalyan. 2022. Numglue: A suite of fundamental yet challenging mathematical reasoning tasks. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 3505–3523. Association for Computational Linguistics.

Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 839–849, San Diego, California. Association for Computational Linguistics.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2022. Crosslingual generalization through multitask finetuning. *Preprint*, arXiv:2211.01786.

Nostalgebraist. 2020. Interpreting gpt: The logit lens. LessWrong.

OpenAI. 2023. GPT-4 technical report. *CoRR*, abs/2303.08774.

OpenAI. 2025. Openai o3 and o4-mini system card.

OpenCompass. 2023. Opencompass: A universal evaluation platform for foundation models. https://github.com/open-compass/opencompass.

Cheonbok Park, Jeonghoon Kim, Joosung Lee, Sanghwan Bae, Jaegul Choo, and Kang Min Yoo. 2025. Cross-lingual collapse: How language-centric foundation models shape reasoning in large language models. *arXiv preprint arXiv:2506.05850*.

Long Phan, Alice Gatti, Ziwen Han, Nathaniel Li, Josephina Hu, Hugh Zhang, Sean Shi, Michael Choi, Anish Agrawal, Arnav Chopra, Adam Khoja, Ryan Kim, Jason Hausenloy, Oliver Zhang, Mantas Mazeika, Daron Anderson, Tung Nguyen, Mobeen Mahmood, Fiona Feng, Steven Y. Feng, Haoran Zhao, Michael Yu, Varun Gangal, Chelsea Zou, Zihan Wang, Jessica P. Wang, Pawan Kumar, Oleksandr Pokutnyi, Robert Gerbicz, Serguei Popov, John-Clark Levin, Mstyslav Kazakov, Johannes Schmitt, Geoff Galgon, Alvaro Sanchez, Yongki Lee, Will Yeadon, Scott Sauers, Marc Roth, Chidozie Agu, Søren Riis, Fabian Giska, Saiteja Utpala, Zachary Giboney, Gashaw M. Goshu, Joan of Arc Xavier, Sarah-Jane Crowson, Mohinder Maheshbhai Naiya, Noah Burns, Lennart Finke, Zerui Cheng, Hyunwoo Park, Francesco Fournier-Facio, John Wydallis, Mark Nandor, Ankit Singh, Tim Gehrunger, Jiaqi Cai, Ben McCarty, Darling Duclosel, Jungbae Nam, Jennifer Zampese, Ryan G. Hoerr, Aras Bacho, Gautier Abou Loume, Abdallah Galal, Hangrui Cao, Alexis C. Garretson, Damien Sileo, Qiuyu Ren, Doru Cojoc, Pavel Arkhipov, Usman Qazi, Lianghui Li, Sumeet Motwani, Christian Schröder de Witt, Edwin Taylor, Johannes Veith, Eric Singer, Taylor D. Hartman, Paolo Rissone, Jaehyeok Jin, Jack Wei Lun Shi, Chris G. Willcocks, Joshua Robinson, Aleksandar Mikov, Ameya Prabhu, Longke Tang, Xavier Alapont, Justine Leon Uro, Kevin Zhou, Emily de Oliveira Santos, Andrey Pupasov Maksimov, Edward Vendrow, Kengo Zenitani, Julien Guillod, Yuqi Li, Joshua Vendrow, Vladyslav Kuchkin, and Ng Ze-An. 2025. Humanity's last exam. *CoRR*, abs/2501.14249.

Edoardo Maria Ponti, Goran Glavaš, Olga Majewska, Qianchu Liu, Ivan Vulić, and Anna Korhonen. 2020. XCOPA: A multilingual dataset for causal commonsense reasoning. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2362–2376, Online. Association for Computational Linguistics.

Jirui Qi, Shan Chen, Zidi Xiong, Raquel Fernández, Danielle Bitterman, and Arianna Bisazza. 2025. When models reason in your language: Controlling thinking language comes at the cost of accuracy. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 20279–20296, Suzhou, China. Association for Computational Linguistics.

Lihua Qian, Hao Zhou, Yu Bao, Mingxuan Wang, Lin Qiu, Weinan Zhang, Yong Yu, and Lei Li. 2021. Glancing transformer for non-autoregressive neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1993–2003.

Samyam Rajbhandari, Jeff Rasley, Olatunji Ruwase, and Yuxiong He. 2020. Zero: memory optimizations toward training trillion parameter models. In *Proceedings of the International Conference for High Performance Computing, Networking, Storage and Analysis, SC 2020, Virtual Event / Atlanta, Georgia, USA, November 9-19, 2020*, page 20. IEEE/ACM.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S. Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *Logical Formalizations of Commonsense Reasoning, Papers from the 2011 AAAI Spring Symposium, Technical Report SS-11-06, Stanford, California, USA, March 21-23, 2011*. AAAI.

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

Shuaijie She, Wei Zou, Shujian Huang, Wenhao Zhu, Xiang Liu, Xiang Geng, and Jiajun Chen. 2024. MAPO: advancing multilingual reasoning through multilingual-alignment-as-preference optimization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 10015–10027. Association for Computational Linguistics.

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.

Taiwei Shi, Yiyang Wu, Linxin Song, Tianyi Zhou, and Jieyu Zhao. 2025. Efficient reinforcement finetuning via adaptive curriculum learning. *arXiv preprint arXiv:2504.05520*.

Alexey Tikhonov and Max Ryabinin. 2021. It's all in the heads: Using attention heads as a baseline for cross-lingual transfer in commonsense reasoning. *Preprint*, arXiv:2106.12066.

Xinyi Wang, Jinyi Han, Zishang Jiang, Tingyun Li, Jiaqing Liang, Sihang Jiang, Zhaoqian Dai, Shuguang Ma, Fei Yu, and Yanghua Xiao. 2025a. Hint: Helping ineffective rollouts navigate towards effectiveness. *arXiv preprint arXiv:2510.09388*.

Yilin Wang, Yuchun Fan, Jiaoyang Li, Ziming Zhu, Yongyu Mu, Qiaozhi He, Tong Xiao, and Jingbo Zhu. 2026. Dapt: A dual-path framework for multilingual multi-hop question answering. *arXiv preprint arXiv:2603.19097*.

Yiming Wang, Pei Zhang, Jialong Tang, Haoran Wei, Baosong Yang, Rui Wang, Chenshu Sun, Feitong Sun, Jiran Zhang, Junxuan Wu, et al. 2025b. Polymath: Evaluating mathematical reasoning in multilingual contexts. *arXiv preprint arXiv:2504.18428*.

Liang Wen, Yunke Cai, Fenrui Xiao, Xin He, Qi An, Zhenyu Duan, Yimin Du, Junchen Liu, Lifu Tang, Xiaowei Lv, Haosheng Zou, Yongchao Deng, Shousheng Jia, and Xiangzheng Zhang. 2025. Light-r1: Curriculum sft, DPO and RL for long COT from scratch and beyond. *CoRR*, abs/2503.10460.

Weihao Xuan, Rui Yang, Heli Qi, Qingcheng Zeng, Yunze Xiao, Aosong Feng, Dairui Liu, Yun Xing, Junjue Wang, Fan Gao, Jinghui Lu, Yuang Jiang, Huitao Li, Xin Li, Kunyu Yu, Ruihai Dong, Shangding Gu, Yuekang Li, Xiaofei Xie, Felix Juefei-Xu, Foutse Khomh, Osamu Yoshie, Qingyu Chen, Douglas Teodoro, Nan Liu, Randy Goebel, Lei Ma, Edison Marrese-Taylor, Shijian Lu, Yusuke Iwasawa, Yutaka Matsuo, and Irene Li. 2025. Mmlu-prox: A multilingual benchmark for advanced large language model evaluation. *arXiv preprint arXiv:2503.10497*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. 2025a. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Wen Yang, Junhong Wu, Chong Li, Chengqing Zong, and Jiajun Zhang. 2025b. Parallel scaling law: Unveiling reasoning generalization through a cross-linguistic perspective. *arXiv preprint arXiv:2510.02272*.

Junchi Yao, Shu Yang, Jianhua Xu, Lijie Hu, Mengdi Li, and Di Wang. 2025. Understanding the repeat curse in large language models from a feature perspective. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 7787–7815, Vienna, Austria. Association for Computational Linguistics.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, YuYue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, LingJun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. 2025. DAPO: An open-source LLM reinforcement learning system at scale. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.

Kaiyi Zhang, Ang Lv, Jinpeng Li, Yongbo Wang, Feng Wang, Haoyuan Hu, and Rui Yan. 2025a. Stephint: Multi-level stepwise hints enhance reinforcement learning to reason. *arXiv preprint arXiv:2507.02841*.

Wen Zhang, Yang Feng, Fandong Meng, Di You, and Qun Liu. 2019. Bridging the gap between training and inference for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4334–

4343, Florence, Italy. Association for Computational Linguistics.

Xue Zhang, Yunlong Liang, Fandong Meng, Songming Zhang, Yufeng Chen, Jinan Xu, and Jie Zhou. 2025b. CM-align: Consistency-based multilingual alignment for large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2025*, pages 25689–25702, Suzhou, China. Association for Computational Linguistics.

Xue Zhang, Yunlong Liang, Fandong Meng, Songming Zhang, Yufeng Chen, Jinan Xu, and Jie Zhou. 2025c. Multilingual knowledge editing with language-agnostic factual neurons. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 5775–5788, Abu Dhabi, UAE. Association for Computational Linguistics.

Xue Zhang, Yunlong Liang, Fandong Meng, Songming Zhang, Kaiyu Huang, Yufeng Chen, Jinan Xu, and Jie Zhou. 2025d. Think natively: Unlocking multilingual reasoning with consistency-enhanced reinforcement learning. *arXiv preprint arXiv:2510.07300*.

Yidan Zhang, Yu Wan, Boyi Deng, Baosong Yang, Hao-Ran Wei, Fei Huang, Bowen Yu, Dayiheng Liu, Junyang Lin, Fei Huang, and Jingren Zhou. 2025e. P-MMEval: A parallel multilingual multitask benchmark for consistent evaluation of LLMs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 4809–4836, Suzhou, China. Association for Computational Linguistics.

Jinman Zhao, Erxue Min, Hui Wu, Ziheng Li, Zexu Sun, Hengyi Cai, Shuaiqiang Wang, Xu Chen, and Gerald Penn. 2026. Beyond step pruning: Information theory based step-level optimization for self-refining large language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 40(41):34941–34949.

Jinman Zhao and Xueyan Zhang. 2024. Large language model is not a (multilingual) compositional relation reasoner. In *First Conference on Language Modeling*.

Yiran Zhao, Wenxuan Zhang, Guizhen Chen, Kenji Kawaguchi, and Lidong Bing. 2024. How do large language models handle multilingualism? *CoRR*, abs/2402.18815.

A Preliminary Study

A.1 Construction of Training Data

Due to the scarcity of multilingual mathematical training data, we construct our training data from DeepMath-103K (He et al., 2025), which contains high-difficulty math problems paired with reasoning traces generated by DeepSeek-R1 (Guo et al., 2025). We translate the questions and their corresponding reasoning traces from the DeepMath-103K dataset (He et al., 2025) into ten in-domain languages: Arabic, Bengali, Thai, Swahili, Japanese, Chinese, German, French, and Russian using GPT-4o-mini (OpenAI, 2023), followed by manual verification and correction.

A.2 Metrics for Preliminary Study

We provide a detailed description of the metrics used in the preliminary study below:

Policy Entropy. The policy entropy can be described as follows. Given a question q and a policy model π_θ, the model generates a response $y = (y_1, ..., y_t, ..., y_T)$, where each token y_t is sampled from the conditional distribution $\pi_\theta(\cdot \mid y_{<t}, q)$. Following Shi et al. (2025), we measure the average token-level entropy of the policy model over the training dataset $\mathcal{D}$, as defined by the following equation:

$$\mathcal{H}(\pi_\theta, \mathcal{D}) = \mathbb{E}_{q\sim\mathcal{D},\, y\sim\pi_\theta(\cdot|q)} \left[\frac{1}{|y|} \sum_{t=1}^{|y|} -\log \pi_\theta(y_t \mid y_{<t}, q) \right]. \tag{1}$$

Average Accuracy. We report the average accuracy over ten languages on the MMATH test sets by comparing the extracted final answer in the response to the gold answer.

Reward Score. We report the average reward score, defined as the proportion of rollouts with $R(o) = 1$, where $R(o)$ equals 1 only when all reward criteria are satisfied, defined as equation 9.

Response Length. We track the model's response length on the MMATH test sets throughout training. For each evaluation checkpoint, we tokenize the generated responses using the model's own tokenizer and report the average number of output tokens (including both the reasoning trace and final answer) across the ten languages.

Repeat Score. Following (Yao et al., 2025)'s work, we measure repetition using the weighted n-gram repetition rate. Before computing repetition, we remove mathematical expressions and symbols from each output via regular expressions and compute the metric on the remaining text. Given a generated token sequence $y = (t_1, \ldots, t_T)$, the contiguous n-gram starting at position j is defined as:

$$n\text{-gram}_j = (t_j, t_{j+1}, \ldots, t_{j+n-1}), \quad j = 1, \ldots, T-n+1. \tag{2}$$

Let $\{n_i\}_{i=1}^{K}$ denote the unique n-grams in y, with frequency f_i for each n_i. We compute the repeat score as specified by the following equation:

$$\text{Repeat Score}_n(y) = \frac{\sum_{i=1}^{K} f_i^w \cdot \mathbb{I}(f_i > 1)}{\sum_{i=1}^{K} \max(f_i, 1)^w}, \tag{3}$$

where $\mathbb{I}(\cdot)$ is the indicator function and w is a weighting factor. Following (Yao et al., 2025)'s setup, we set $n = 1$, $w = 1$ and report the average repeat score over all generated outputs across ten languages on the MMATH benchmark.

B Experimental Details of Multilingual Mathematical Reasoning Task

B.1 Evaluation Datasets

The datasets for evaluating the multilingual reasoning ability of LLMs cover two benchmarks: MMATH and PolyMath. The detailed descriptions of these two datasets are as follows.

MMATH (Luo et al., 2025) is a new benchmark for evaluating multilingual complex reasoning. It contains 374 high-quality math problems spanning 10 typologically and geographically diverse languages, resulting in 3,740 test instances in total. The source problems are drawn from 30 questions from AIME 2024, 15 from AIME 2025[6], 18 from CNMO[7], and 311 filtered problems from MATH500 (Lightman et al., 2024). Each problem is translated and validated through a rigorous pipeline that combines frontier LLMs with human verification, ensuring semantic equivalence across languages. The test sets include 10 languages: English (en), Chinese (zh), Arabic (ar), Spanish (es), French (fr), Japanese (ja), Korean (ko), Portuguese (pt), Thai (th), and Vietnamese (vi).

PolyMath (Wang et al., 2025b) is a multilingual benchmark organized by comprehensive difficulty levels, spanning from K-12 to Olympiad and advanced frontier mathematics. It covers 18 languages: English (en), Chinese (zh), Spanish (es), Arabic (ar), French (fr), Bengali (bn), Portuguese (pt), Russian (ru), Indonesian (id), German (de), Japanese (ja), Swahili (sw), Vietnamese (vi), Italian (it), Telugu (te), Korean (ko), Thai (th), and Malay (ms). And the problems are categorized into four difficulty levels based on Thought Depth and Knowledge Breadth, with 125 problems per level. The data sources for each level are summarized as follows:

- **Low-level:** Problems are sourced from MGSM (Shi et al., 2023), a multilingual math word-problem benchmark, with additional translations supplemented by P-MMeval (Zhang et al., 2025e).
- **Medium-level:** Problems are collected from *College Math* exercise sets and standardized examinations (e.g., China's *Gaokao* and postgraduate entrance exams), together with problems from widely used contest archives such as the U.S. AMC and China's provincial CNMO selection contests.
- **High-level:** Problems are drawn from established competition problem sets, including the U.S. AIME and China's CNMO.
- **Top-level:** Problems are aggregated from the IMO/IMO Shortlist and major national/regional Olympiads (e.g., CMO, USAMO, Putnam), and complemented with frontier problems from the HLE dataset (Phan et al., 2025).

B.2 Evaluation Details

B.2.1 Evaluation Details for MMATH

Following Luo et al. (2025), we generate outputs using a temperature of t = 0.6, a top-p value of 0.95, and a maximum output length of 32,768 tokens. To obtain a more reliable estimate of reasoning accuracy, each evaluation is repeated 4 times, and the average result is recorded. Given the varying complexity of each benchmark subset, we report the final score using macro-average accuracy. To extract the final answer, we employ the math extraction tool from OpenCompass (2023), and the extracted answers are then verified against the ground truth using `math_verify`[8]. We utilize the

[6] https://maa.org/maa-invitational-competitions/
[7] https://www.cms.org.cn/
[8] https://github.com/huggingface/Math-Verify

inference prompt from Luo et al. (2025). The evaluation prompts used for different methods on the MMATH test set are shown in Figure 9.

B.2.2 Evaluation Details for PolyMath

For a fair comparison, we use greedy decoding to ensure the determinism of the outputs and set the maximum generation length to 65,536 tokens during inference. To ensure a fair comparison and mitigate the risk of hallucinations (Huang et al., 2025c, 2024a, 2025b; Chen et al., 2025; Zhao et al., 2026; Huang et al., 2025a, 2024b), we adopt the inference prompt from Luo et al. (2025). The evaluation prompts used for different methods on the PolyMath test set are shown in Figure 10.

Following Luo et al. (2025), we report the Difficulty-Weighted Accuracy (DW-ACC) (Luo et al., 2025) on the PolyMath benchmark. DW-ACC assigns level-specific weights w_1, w_2, w_3, w_4 to problems from the *low*, *medium*, *high*, and *top* levels, respectively, with weights doubling as difficulty increases: $w_1 = 1$, $w_2 = 2$, $w_3 = 4$, and $w_4 = 8$. This weighting scheme reduces the influence of easier problems and places greater emphasis on correctness at higher difficulty levels. The accuracies at the four levels are denoted by a_1, a_2, a_3, a_4, and DW-ACC is specified by the following equation:

$$\text{DW-ACC} = \frac{\sum_{i=1}^{4} w_i a_i}{\sum_{i=1}^{4} w_i} = \sum_{i=1}^{4} \left(\frac{2^{i-1}}{15} a_i \right). \quad (4)$$

C Experimental Details of LANG

C.1 Training Datasets

We train our models on the multilingual DeepMath-103K dataset, with full details of the data construction provided in Appendix A.1.

C.2 Training Details of Cold-Start training

We utilize LLaMA-Factory[9] as our cold-start training framework. All models are trained using NVIDIA H200 GPUs. All models are trained for 1 epoch with a batch size of 256, and the learning rate is set to 2e-5. We set the maximum token length to be 16384. We use Deepspeed stage 2 (Rajbhandari et al., 2020) to conduct multi-GPU distributed training, with training precision Bfloat16 enabled.

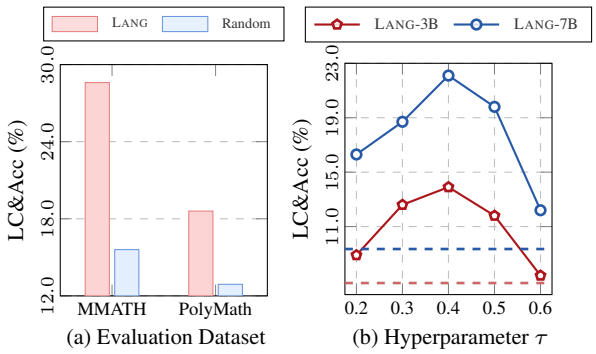

Figure 8: (a) The impact of randomly discarding segments of the hint on model performance. (b) The effect of the choice of the hyperparameter τ on the performance of LANG. Red dashed line denotes the performance of Qwen2.5-3B-Instruct, and Blue dashed line denotes the performance of Qwen2.5-7B-Instruct.

C.3 Training Details of RL training

We utilize the GRPO algorithm implemented by verl[10]. All models are trained using 4×8 NVIDIA H200 GPUs. All models are trained for 5 epochs with a batch size of 128 and a PPO mini-batch size of 64. We use a learning rate of 1×10^{-6}, 8 rollouts with temperature 1.0, and a KL coefficient of 0.0. The maximum sequence length is 16,384 tokens. Following Joshi et al. (2020), the languages in this study are classified into three categories based on resource availability: high-resource languages include English, German, French, Spanish, Portuguese, and Italian; mid-resource languages include Japanese, Chinese, Russian, Korean, and Vietnamese; and low-resource languages include Arabic, Bengali, Thai, Swahili, Telugu, and Indonesian.

D Detailed Results for LCR and Accuracy

We report the LCR of Qwen2.5-3B-Instruct on MMATH and PolyMath in Figure 11 and Figure 13, respectively. For Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct on the PolyMath test sets, we report the difficulty-weighted accuracy in Table 11, and accuracy by difficulty level in Tables 12–15. For the PolyMath test sets, we report the difficulty-weighted accuracy in Table 11. We additionally report per-tier accuracies for the four difficulty levels in Tables 12–15.

[9] https://github.com/hiyouga/LLaMA-Factory

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

E Additional Ablation Studies

Effect of random hint segment discarding. To evaluate the necessity of truncating hints from the tail, we conduct ablation studies by maintaining the total length of hint decay, but randomly discarding segments of the hint. As shown in Figure 8 (a), randomly discarding segments causes semantic disruption in the multilingual hints, leading to a sharp decline in the model's average performance across both benchmarks. This suggests that semantically disrupted multilingual hints fail to provide effective guidance during training, making it difficult for the model to learn coherent and logical reasoning traces.

Effect of τ reveals a meaningful trade-off across language groups yet consistent robustness. To demonstrate the impact of the choice of τ on our method, we conduct ablation studies by training Qwen2.5-3B-Instruct and Qwen2.5-7B-Instruct with different values of τ, reporting their average performance on MMATH and PolyMath. As shown in Figure 8 (b), selecting too small a value for τ may cause hints to be turned off too early during training, preventing the model from receiving sufficient multilingual hint guidance and making it difficult to explore correct reasoning traces. In contrast, choosing too large a value for τ may cause hints to be turned off too late during training, leading the model to become overly reliant on hint guidance and preventing it from developing sufficient autonomous exploration capabilities. More concretely, the hyperparameter τ mediates a meaningful trade-off across language groups with different resource levels. A smaller τ removes hints earlier, which benefits high-resource languages by encouraging more autonomous exploration, but deprives low-resource languages of the guidance they critically need. A larger τ retains hints longer, which provides low-resource languages with extended support but may induce dependency in high-resource ones. To illustrate this, we report the performance of Qwen2.5-7B-Instruct on PolyMath across high-, medium-, and low-resource language groups in Table 5. As shown, $\tau = 0.4$ achieves the best overall balance across all three groups, while $\tau = 0.3$ favors high-resource languages at the cost of low-resource performance. The sensitivity observed in Figure 8 (b) thus reflects the inherent difficulty gap across language groups rather than a design fragility. Notably, our method consistently outperforms the Qwen2.5-3B/7B-Instruct baselines regardless of the value of τ. Even at the suboptimal setting of $\tau = 0.3$, LANG still surpasses LC-GRPO on low-resource languages by 7.7%, demonstrating strong robustness precisely where improvement is most needed. These results highlight that LANG can robustly enhance multilingual reasoning ability by injecting multilingual hints during the early stages of training to guide correct reasoning path generation, while adaptively removing hints to encourage autonomous exploration.

	High-Resource	Medium-Resource	Low-Resource
LANG (τ=0.3)	17.8	20.2	18.1
LANG (τ=0.4)	25.1	21.5	19.6
LC-GRPO	23.2	20.0	16.8

Table 5: Performance of Qwen2.5-7B-Instruct on PolyMath across high-, medium-, and low-resource language groups under different τ values.

F Comprehensive Results for Additional Training Models

We additionally report LC&Acc results for Qwen2.5-32B-Instruct and Llama3.1-8B-Instruct. Table 8 summarizes LC&Acc on the MMATH test set, and Table 9 reports LC&Acc on the PolyMath test set.

G Experimental Details of Multilingual Non-Mathematical Reasoning Tasks

To further validate the effectiveness and generalizability of our method, we evaluate it on multilingual understanding benchmarks, including MMLU-ProX (Xuan et al., 2025), XWinograd (Muennighoff et al., 2022; Tikhonov and Ryabinin, 2021), and XCOPA (Ponti et al., 2020), and additionally assess its performance on the multilingual generation benchmark XStoryCloze (Lin et al., 2021). For a fair comparison, we utilize lm-evaluation-harness[11] as our evaluation framework. The details of the four benchmarks are summarized as follows:

MMLU-ProX (Xuan et al., 2025) is a novel multilingual benchmark that builds upon the challenging, reasoning-focused design of MMLU-Pro while extending its coverage to 29 typologically diverse languages, including English (en), Chinese (zh), Japanese (ja), Korean (ko), French (fr), German (de), Spanish (es), Portuguese (pt), Arabic (ar), Thai (th), Hindi (hi), Bengali (bn), Swahili (sw),

[11] https://github.com/EleutherAI/lm-evaluation-harness

Afrikaans (af), Czech (cs), Hungarian (hu), Indonesian (id), Italian (it), Marathi (mr), Nepali (ne), Russian (ru), Serbian (sr), Telugu (te), Ukrainian (uk), Urdu (ur), Vietnamese (vi), Wolof (wo), Yoruba (yo), and Zulu (zu). MMLU-ProX provides 11,829 parallel questions aligned across these languages, thereby enabling a comprehensive evaluation of LLMs' multilingual reasoning capabilities.

XWinograd (Muennighoff et al., 2022; Tikhonov and Ryabinin, 2021) extends the original English Winograd Schema Challenge (WSC) (Levesque et al., 2012) to five additional languages: French (fr), Japanese (ja), Portuguese (pt), Russian (ru), and Chinese (zh). The dataset comprises pronoun-resolution problems designed to evaluate a model's commonsense reasoning ability.

XCOPA (Ponti et al., 2020) is a multilingual extension of the Choice of Plausible Alternatives (COPA) task, constructed by translating and re-annotating the validation and test sets of the English (en) COPA dataset (Roemmele et al., 2011) into 11 target languages, including Estonian (et), Haitian Creole (ht), Indonesian (id), Italian (it), Quechua (qu), Swahili (sw), Tamil (ta), Thai (th), Turkish (tr), Vietnamese (vi), and Chinese (zh). Each instance consists of a premise, a question (cause or result), and two candidate alternatives. The task is to select the more plausible alternative.

XStoryCloze (Lin et al., 2021) is constructed by Lin et al. (2022) by translating the validation split of the original English StoryCloze dataset (Mostafazadeh et al., 2016) into 10 typologically diverse languages: Russian (ru), Chinese (zh), Spanish (es), Arabic (ar), Hindi (hi), Indonesian (id), Telugu (te), Swahili (sw), Basque (eu), and Burmese (my). Each example consists of a four-sentence commonsense story, a correct ending, and a plausible but incorrect ending.

H Experimental Details of Baseline Methods

We conduct experiments on the Qwen2.5-3B-Instruct, Qwen2.5-7B-Instruct, Qwen2.5-32B-Instruct (Yang et al., 2025a) and Llama3.1-8B-Instruct (AI@Meta, 2024). The detailed baseline implementation is as follows:

H.1 Prompting-based Methods

The evaluation prompt templates of prompting-based methods used for MMATH and PolyMath datasets are shown in Figure 9 and 10, respectively. In the template, {*input*} can be replaced with multilingual questions.

Language-Constraint Prompting (LCP) (Wang et al., 2025b): For each question in the MMATH and PolyMath test sets, LCP explicitly prompts models to solve questions in targeted languages.

Discourse-Initiated Thinking (DIT) (Luo et al., 2025): Inspired by the phenomenon, the model tends to start thinking with discourse markers like "Alright" or "Okay". DIT extracts these multilingual markers from native prompt responses and append after the "<think>" token. This approach encourages models to initiate their reasoning using discourse cues as entry points into the multilingual thinking process.

Question-Restatement Thinking (QRT) (Luo et al., 2025): Inspired by another common phenomenon, which models often restate the question before engaging in actual reasoning. QRT replicates this behavior by explicitly inserting a restated version of the question at the beginning of the thinking process. This intervention encourages the model to frame the problem before attempting to solve it.

H.2 Training-based Methods

Multilingual Supervised Fine-tuning (M-SFT): This method involves directly full-parameter fine-tuning models with constructed multilingual data. During training, we train all models for 3 epochs with a batch size of 256, and the learning rate is maintained at 2e-5 using NVIDIA H200 GPUs.

Vanilla GRPO (Guo et al., 2025): We adopt the vanilla GRPO algorithm with only format and accuracy rewards to enhance the model's multilingual reasoning ability. We use the same training data as in LANG.

Language-Consistency GRPO (LC-GRPO): To maintain input–output language consistency, we further incorporate the language consistency reward into the GRPO algorithm. To ensure a fair comparison, we use the same training hyperparameters as LANG.

M-Thinker (Zhang et al., 2025d): Following Zhang et al. (2025d), we use the Light-R1-SFTData dataset (Wen et al., 2025), which contains approximately 76K samples. Each sample consists of an English question paired with a high-quality response generated by DeepSeek-R1 (DeepSeek-AI, 2025). We then use `DeepSeek-V3-0324` to translate the English questions into ten in-domain languages: Arabic, Bengali, Thai, Swahili, Japanese, Chinese, German, French, and Russian.

For cold-start training, we randomly sample 7.5K questions for each in-domain language and employ `DeepSeek-R1-0528` to generate responses in the same language as the input, yielding 75K cold-start samples in total. For RL training, we perform rejection sampling on the remaining Light-R1-SFTData with $N = 8$ sampled candidates per prompt. We keep a prompt only if it yields mixed outcomes among the candidates (i.e., $0 < |O_{\text{correct}}| < N$), which avoids degenerate groups with all-zero or all-one rewards and thus provides effective advantage signals for GRPO. From the resulting filtered RL pool, we randomly select 3K samples per in-domain language to ensure a language-balanced training set under a fixed compute budget. We use `DeepSeek-V3-0324` to compute the alignment ratio by measuring the overlap between the English reasoning trace and the corresponding target-language reasoning trace, and we set the maximum generation length to 16,384 tokens.

mGRPO (mGRPO, 2025): This method proposes a Polyglot Reasoning Generation Module (PRGM) to guide the LLM to generate a group of n multilingual responses for each question. Given an input question, we produce n candidate responses using prompts with or without explicit language instructions. Specifically, one response is generated without any language constraint, while the remaining responses are generated with prompts that explicitly specify a reasoning language randomly sampled from a predefined multilingual set, thereby encouraging broader exploration of the multilingual reasoning space. Following mGRPO (2025), we adopt the mathematical reasoning dataset from MAPO (She et al., 2024) as training data, which contains 1,703 English questions from a subset of NumGLUE along with ChatGPT-translated versions in nine languages. Since the MAPO training data does not include Arabic translations, we additionally translate the NumGLUE (Mishra et al., 2022) questions into Arabic using GPT-4o-mini. For the Qwen2.5 series models, we use 10 in-domain languages set to guide the rollout, matching the languages covered by the training data. We set the learning rate to 1e-6, sample 5 rollouts per prompt, and train for 5 epochs with a batch size of 128.

Method	Ar	Th	Fr	Ja	Zh	En	Vi	Ko	Pt	Es	Avg.
Qwen2.5-7B-Instruct	0.3	0.5	0.2	3.6	21.0	28.2	1.3	1.8	0.7	2.1	6.0
+ LANG	26.1	27.8	31.8	28.2	21.4	33.1	29.5	23.8	32.6	31.1	28.5

Table 6: The LC&Acc (%) on MMATH under different teacher models for reasoning-trace generation, using Qwen2.5-7B-Instruct as the backbone.

Method	Ar	Bn	Th	Sw	Ja	Zh
Qwen2.5-7B-Instruct	12.1	11.8	12.7	2.2	13.8	15.3
+ LANG	16.0	10.6	15.5	3.6	14.9	15.8
Method	**De**	**Fr**	**Ru**	**En**	**Te**	**Ko**
Qwen2.5-7B-Instruct	15.3	15.2	14.8	17.3	6.2	12.4
+ LANG	17.2	17.4	19.7	18.0	7.7	15.8
Method	**Vi**	**It**	**Id**	**Pt**	**Es**	**Avg.**
Qwen2.5-7B-Instruct	14.4	13.4	9.7	13.6	15.1	12.7
+ LANG	17.3	19.8	17.2	17.9	18.8	15.5

Table 7: The LC&Acc (%) on PolyMath under different teacher models for reasoning-trace generation, using Qwen2.5-7B-Instruct as the backbone.

Method	In-Domain Languages							Out-of-Domain Languages					ALL-Avg.
	Ar	Th	Fr	Ja	Zh	En	Avg.	Vi	Ko	Pt	Es	Avg.	
Llama3.1-8B-Instruct	7.2	9.7	13.0	7.9	9.3	15.2	10.4	10.8	8.4	11.2	12.4	10.7	10.5
	7.4	10.0	13.1	9.3	11.0	15.2	11.0	10.8	8.9	11.7	12.7	11.0	11.0
+ Vanilla GRPO	3.9	7.0	6.0	3.5	2.6	9.7	5.5	6.2	3.3	5.5	6.9	5.5	5.5
+ LC-GRPO	3.4	8.8	6.9	**8.8**	6.2	5.8	6.6	7.0	4.9	6.2	7.0	6.3	6.5
+ LANG	**7.4**	**9.9**	**14.3**	8.6	**12.2**	**16.7**	**11.5**	**11.1**	**9.3**	**12.4**	**12.9**	**11.4**	**11.5**
Qwen2.5-32B-Instruct	26.2	26.1	27.9	27.1	24.6	34.7	27.8	27.6	25.8	30.8	29.6	28.4	28.0
	26.8	26.6	27.9	27.1	26.7	34.7	28.3	28.0	26.3	30.9	29.6	28.7	28.5
+ Vanilla GRPO	10.0	16.5	39.6	35.9	25.0	39.5	27.8	29.1	1.1	20.5	18.6	17.3	23.6
+ LC-GRPO	36.7	**38.2**	39.5	37.4	35.4	40.3	37.9	**40.3**	**39.1**	39.9	39.3	39.6	38.6
+ LANG	**37.4**	37.0	**43.5**	**44.1**	**38.5**	**42.5**	**40.5**	39.8	38.2	**42.7**	**42.8**	**40.9**	**40.6**

Table 8: The LC&Acc (%) on MMATH test sets for Llama3.1-8B-Instruct and Qwen2.5-32B-Instruct.

Method	In-Domain Languages											Out-of-Domain Languages								ALL-Avg.
	Ar	Bn	Th	Sw	Ja	Zh	De	Fr	Ru	En	Avg.	Te	Ko	Vi	It	Id	Pt	Es	Avg.	
Llama3.1-8B-Instruct	5.9	5.6	5.9	5.1	6.9	4.3	5.8	7.0	6.3	10.3	6.3	5.0	6.2	7.4	7.4	6.3	5.8	6.3	6.3	6.3
	5.9	6.5	7.1	6.3	6.9	7.1	6.6	7.7	7.3	10.3	7.2	5.2	6.8	7.5	8.1	7.1	6.3	6.3	6.8	7.0
+ Vanilla GRPO	**8.3**	6.2	7.7	6.1	7.1	**5.9**	8.4	9.6	9.0	10.6	7.9	**6.6**	6.5	8.1	8.9	7.4	9.0	9.7	8.0	7.9
+ LC-GRPO	3.0	5.8	7.3	4.7	7.4	4.7	**9.7**	5.1	8.5	7.8	6.4	5.5	6.2	4.6	4.0	2.9	5.3	6.0	4.9	5.8
+ LANG	8.2	**7.3**	**7.8**	**6.5**	**8.9**	5.1	9.6	**10.9**	**11.0**	**13.4**	**8.9**	5.6	**6.9**	**8.5**	**9.7**	**9.3**	**10.8**	**9.8**	**8.7**	**8.8**
Qwen2.5-32B-Instruct	19.8	16.3	16.2	9.4	17.7	20.4	19.0	18.3	19.0	20.5	17.6	10.8	19.0	18.7	20.9	19.0	18.1	18.1	17.8	17.7
	20.3	17.0	6.2	11.9	17.7	21.8	19.1	18.6	19.6	20.5	18.3	11.8	19.0	19.2	20.9	19.0	21.0	18.5	18.5	18.4
+ Vanilla GRPO	15.4	3.5	10.9	3.0	**22.3**	**22.3**	20.2	22.0	**24.3**	20.9	16.5	8.3	17.5	19.4	22.8	21.8	**23.0**	**21.9**	19.2	17.6
+ LC-GRPO	19.4	19.8	**20.9**	9.9	21.5	21.8	22.0	19.4	21.4	**21.5**	19.8	13.4	21.3	**23.5**	21.0	21.6	21.7	20.6	20.4	20.0
+ LANG	**19.9**	**20.3**	19.3	**13.2**	21.4	22.2	**23.4**	**22.3**	22.9	21.4	**20.6**	**14.9**	**22.0**	22.7	**23.8**	**22.8**	21.2	20.9	**21.2**	**20.9**

Table 9: The LC&Acc (%) on PolyMath test sets for Llama3.1-8B-Instruct and Qwen2.5-32B-Instruct.

Method	In-Domain Languages							Out-of-Domain Languages					ALL-Avg.
	Ar	Th	Fr	Ja	Zh	En	Avg.	Vi	Ko	Pt	Es	Avg.	
Qwen2.5-3B-Instruct	13.1	11.0	16.7	13.7	15.1	23.0	15.4	15.9	13.0	18.5	16.6	16.0	15.7
+ LCP	13.7	13.3	16.5	12.5	15.9	21.3	15.5	13.1	11.0	16.0	16.7	14.2	15.0
+ DIT	12.6	12.6	17.0	12.7	17.4	**23.2**	15.9	14.9	12.5	17.8	17.9	15.8	15.8
+ QRT	12.4	11.4	17.2	13.2	17.9	22.3	15.7	14.1	12.3	16.8	18.6	15.5	15.6
+ M-SFT	12.9	12.1	18.5	13.3	17.0	22.4	16.0	13.4	12.1	16.5	19.1	15.3	15.7
+ Vanilla GRPO	**18.6**	**15.7**	19.5	**18.5**	19.4	22.4	**19.0**	16.0	**20.4**	20.1	19.1	**18.9**	**19.0**
+ LC-GRPO	16.2	15.5	17.8	16.3	17.5	22.9	17.7	16.4	15.5	20.1	19.5	17.9	17.8
+ M-Thinker	12.8	11.6	19.7	11.9	15.5	22.4	15.7	15.0	12.2	17.2	17.7	15.5	15.6
+ mGRPO	15.0	15.3	18.2	12.2	17.1	19.9	16.3	16.3	14.5	18.6	16.7	16.5	16.4
+ LANG	15.1	14.8	**20.7**	15.5	**20.2**	23.1	18.2	**19.6**	15.1	**20.9**	**19.6**	18.8	18.5
Qwen2.5-7B-Instruct	20.3	21.5	24.1	25.4	23.8	28.2	23.9	20.8	24.6	24.1	25.1	23.6	23.8
+ LCP	18.8	19.1	23.2	18.8	22.4	29.6	22.0	19.9	21.0	23.6	23.0	21.9	21.9
+ DIT	21.2	19.9	26.4	23.6	23.0	30.0	24.0	21.6	23.1	24.3	25.5	23.6	23.9
+ QRT	19.1	18.7	24.7	20.4	26.1	27.8	22.8	23.0	19.6	23.7	24.0	22.6	22.7
+ M-SFT	22.7	21.0	22.2	22.1	23.9	**32.6**	24.1	21.7	21.2	22.9	25.3	22.8	23.6
+ Vanilla GRPO	**27.1**	27.4	**31.6**	25.5	28.3	31.1	28.5	28.7	**31.3**	27.9	30.2	29.5	28.9
+ LC-GRPO	24.3	23.6	28.2	25.0	28.2	30.2	26.6	25.2	25.0	29.2	27.9	26.9	26.7
+ M-Thinker	21.4	20.7	22.7	24.5	25.7	28.6	23.9	24.6	22.8	26.7	26.7	25.2	24.4
+ mGRPO	24.7	20.5	27.8	26.5	26.0	31.0	26.1	25.9	25.0	29.9	28.6	27.3	26.6
+ LANG	26.5	**28.6**	31.2	**28.2**	**28.4**	32.1	**29.2**	**30.1**	25.2	**32.4**	**31.3**	**29.8**	**29.4**

Table 10: The accuracy (%) on MMATH test sets.

Method	In-Domain Languages											Out-of-Domain Languages									ALL-Avg.
	Ar	Bn	Th	Sw	Ja	Zh	De	Fr	Ru	En	Avg.	Te	Ko	Ms	Vi	It	Id	Pt	Es	Avg.	
Qwen2.5-3B-Instruct	8.4	4.5	7.6	1.3	8.6	9.4	9.3	9.6	8.9	11.4	7.9	1.0	8.4	7.1	9.9	10.0	9.7	11.8	11.4	8.7	8.2
+ LCP	8.2	3.7	8.4	**3.4**	6.8	11.0	9.9	9.3	8.9	**14.1**	8.4	2.6	7.4	6.3	9.2	9.6	9.8	11.1	9.9	8.2	8.3
+ DIT	7.6	5.4	6.4	3.1	7.4	10.4	**11.7**	9.9	10.4	11.7	8.4	1.0	8.2	9.4	8.8	9.6	9.4	10.5	10.2	8.4	8.4
+ QRT	**10.8**	3.2	6.4	1.6	9.2	7.7	9.6	11.0	10.8	12.7	8.3	1.6	7.5	8.7	11.4	10.7	10.5	9.7	9.7	8.7	8.5
+ M-SFT	7.0	3.5	7.9	2.1	5.7	10.6	8.8	10.2	7.7	13.8	7.7	1.4	8.3	8.6	9.5	10.8	8.9	10.6	10.6	8.6	8.1
+ Vanilla GRPO	9.1	**10.1**	8.1	2.8	8.5	12.1	10.5	11.2	10.6	12.7	9.6	**4.6**	**10.3**	10.2	9.7	8.1	**12.2**	10.6	**11.6**	9.7	9.6
+ LC-GRPO	9.9	6.6	10.6	3.0	9.7	11.8	9.9	11.7	10.2	12.4	9.6	3.0	8.5	9.3	10.5	10.9	12.0	11.5	11.0	9.6	9.6
+ M-Thinker	8.3	4.3	7.7	1.4	7.6	12.0	8.5	8.6	9.9	12.9	8.1	2.8	8.5	**10.5**	10.1	8.9	8.6	9.7	9.3	8.6	8.3
+ mGRPO	8.9	5.9	8.7	1.5	8.3	10.4	10.0	**11.9**	9.1	12.7	8.7	3.1	8.0	8.2	8.6	10.3	8.0	8.7	10.2	8.1	8.5
+ LANG	8.9	7.4	**10.8**	2.7	**9.9**	**15.6**	10.9	11.2	**11.6**	13.3	**10.2**	4.5	9.6	9.4	**12.2**	**11.8**	10.5	**12.3**	11.0	**10.2**	**10.2**
Qwen2.5-7B-Instruct	12.5	12.0	13.3	5.0	13.8	17.4	16.2	15.4	15.7	17.3	13.9	6.2	12.5	13.8	14.5	13.4	13.9	14.8	15.7	13.1	13.5
+ LCP	9.1	5.1	8.3	1.3	8.4	8.3	8.9	9.3	10.9	14.2	8.4	1.0	9.4	10.0	9.2	10.1	8.0	10.7	10.7	8.6	8.5
+ DIT	12.5	10.8	11.8	3.7	13.2	16.1	13.6	16.7	15.8	16.6	13.1	6.6	11.6	10.9	11.9	16.8	13.5	14.9	14.4	12.6	12.9
+ QRT	12.8	11.5	12.2	4.9	14.5	15.6	14.9	16.1	15.1	17.5	13.5	7.7	12.3	11.4	14.0	13.6	13.7	15.1	16.5	13.0	13.3
+ M-SFT	4.5	4.2	6.7	1.7	4.8	4.2	4.1	5.6	4.4	6.1	4.6	1.4	6.6	4.0	6.5	7.1	4.9	7.5	4.7	5.3	4.9
+ Vanilla GRPO	14.0	**15.7**	14.7	6.5	13.6	**18.4**	16.6	15.9	15.7	17.4	14.8	4.3	14.5	14.9	15.9	15.7	14.2	17.1	16.2	14.1	14.5
+ LC-GRPO	14.3	9.0	11.2	7.7	13.7	15.9	17.2	17.0	18.5	17.6	14.2	6.5	13.4	15.0	16.5	16.6	14.5	17.2	18.8	14.8	14.5
+ M-Thinker	13.6	8.8	10.7	5.9	12.0	13.9	13.6	15.3	14.7	15.1	12.3	7.8	11.6	13.1	12.3	14.9	13.7	14.5	15.7	13.0	12.6
+ mGRPO	**16.6**	15.5	13.1	**9.8**	14.7	16.5	16.4	16.1	16.8	16.0	15.2	**9.9**	14.2	13.7	16.1	17.0	16.0	16.4	16.9	15.0	15.1
+ LANG	15.7	11.4	**15.2**	4.1	**15.4**	17.9	**17.9**	**18.0**	**19.6**	**17.9**	**15.3**	7.5	**16.1**	**18.0**	**17.2**	**20.4**	**17.4**	**17.4**	**19.0**	**16.6**	**15.9**

Table 11: The accuracy (%) on PolyMath test sets.

Method	In-Domain Languages											Out-of-Domain Languages									ALL-Avg.
	Ar	Bn	Th	Sw	Ja	Zh	De	Fr	Ru	En	Avg.	Te	Ko	Ms	Vi	It	Id	Pt	Es	Avg.	
Qwen2.5-3B-Instruct	3.2	2.4	2.4	0.8	5.6	3.2	3.2	4.8	2.4	1.6	3.0	0.0	2.4	2.4	4.8	4.8	4.0	**8.0**	5.6	4.0	3.4
+ LCP	3.2	0.8	4.8	3.2	2.4	5.6	4.8	2.4	1.6	**6.4**	3.5	2.4	4.0	0.0	4.0	4.0	4.8	6.4	2.4	3.5	3.5
+ DIT	3.2	4.0	2.4	**3.2**	1.6	4.8	**6.4**	4.8	4.8	4.0	3.9	0.0	4.0	4.8	2.4	1.6	4.0	4.8	4.0	3.2	3.6
+ QRT	**6.4**	1.6	2.4	0.8	4.8	1.6	5.6	4.8	**6.4**	4.0	3.8	0.8	0.8	3.2	6.4	6.4	4.8	3.2	3.2	3.6	3.7
+ M-SFT	1.6	0.8	3.2	2.4	1.6	4.8	3.2	4.8	0.8	5.6	2.9	0.8	4.0	4.8	4.8	**6.4**	3.2	5.6	5.6	4.4	3.6
+ Vanilla GRPO	4.0	4.8	2.4	0.0	4.0	6.4	3.2	4.8	3.2	4.8	3.8	2.4	**5.6**	5.6	4.0	0.8	**7.2**	4.8	**5.6**	**4.5**	4.1
+ LC-GRPO	5.6	3.2	4.8	2.4	4.8	6.4	3.2	4.8	3.2	4.0	4.2	2.4	1.6	3.2	4.0	4.8	6.4	4.8	3.2	3.8	4.0
+ M-Thinker	3.0	1.0	3.4	1.0	2.0	5.8	2.8	3.2	2.8	4.4	2.9	2.2	2.8	**5.8**	5.0	1.4	2.4	3.8	2.8	3.3	3.1
+ mGRPO	4.6	2.0	**5.2**	1.2	1.8	3.4	5.2	**6.6**	3.0	5.0	3.8	2.4	2.2	3.4	1.8	3.0	2.8	1.2	4.6	2.7	3.3
+ LANG	4.0	**4.8**	4.8	2.4	**5.6**	**10.4**	4.8	4.0	5.6	4.8	**5.1**	**3.2**	4.0	2.4	**7.2**	4.0	3.2	5.6	4.0	4.2	**4.7**
Qwen2.5-7B-Instruct	5.6	6.4	6.4	2.4	7.2	**11.2**	11.2	8.0	6.4	**8.8**	7.4	2.4	6.4	6.4	8.0	5.6	4.8	6.4	8.8	6.1	6.8
+ LCP	5.6	1.6	3.2	0.8	4.8	2.4	4.0	3.2	5.6	7.2	3.8	0.0	3.2	5.6	3.2	3.2	1.6	5.6	3.2	3.2	3.6
+ DIT	6.4	5.6	4.8	0.8	**8.0**	10.4	5.6	7.2	8.8	7.2	6.5	4.0	4.8	4.0	4.8	10.4	5.6	6.4	6.4	5.8	6.2
+ QRT	4.8	8.0	5.6	4.8	8.0	8.8	8.8	**9.6**	8.0	8.0	7.4	**5.6**	4.0	2.4	6.4	5.6	6.4	8.8	8.8	6.0	6.8
+ M-SFT	0.8	0.0	1.6	0.8	0.8	0.8	0.0	0.8	0.0	0.0	0.6	0.8	1.6	0.8	0.8	0.0	1.6	0.8	0.8	0.9	0.7
+ Vanilla GRPO	4.8	8.8	**7.2**	3.2	5.6	11.2	8.0	7.2	6.4	7.2	7.0	0.0	5.6	6.4	8.0	6.4	5.6	8.8	8.0	6.1	6.6
+ LC-GRPO	8.0	1.6	3.2	4.0	5.6	8.8	9.6	8.8	12.0	8.0	7.0	2.4	4.0	7.2	8.0	8.8	4.8	8.8	**11.2**	6.9	6.9
+ M-Thinker	6.8	3.2	4.4	3.0	6.8	6.6	7.2	7.0	7.4	6.0	5.8	3.8	4.2	7.4	5.2	7.2	5.8	6.8	7.8	6.0	5.9
+ mGRPO	7.2	**9.6**	4.8	**5.6**	6.4	8.0	8.8	7.2	8.8	6.4	7.3	4.8	5.6	4.8	8.8	7.2	7.2	8.0	8.0	6.8	7.1
+ LANG	**8.0**	4.0	6.4	1.6	7.2	10.4	**11.2**	8.0	**12.0**	8.0	**7.7**	3.2	**6.4**	**10.4**	**9.6**	**12.0**	**10.4**	**9.6**	10.4	**9.0**	**8.3**

Table 12: The accuracy (%) on PolyMath test sets of top-level difficulty.

Method	In-Domain Languages											Out-of-Domain Languages									ALL-Avg.
	Ar	Bn	Th	Sw	Ja	Zh	De	Fr	Ru	En	Avg.	Te	Ko	Ms	Vi	It	Id	Pt	Es	Avg.	
Qwen2.5-3B-Instruct	3.2	0.8	4.8	0.8	4.0	3.2	6.4	3.2	5.6	8.0	4.0	0.8	6.4	2.4	4.8	4.0	4.8	2.4	4.8	3.0	3.9
+ LCP	3.2	1.6	4.8	2.4	4.0	5.6	6.4	7.2	5.6	8.0	4.9	1.6	2.4	3.2	3.2	4.0	4.8	3.2	6.4	3.6	4.3
+ DIT	4.0	0.0	1.6	2.4	6.4	4.8	8.8	4.0	4.0	4.8	4.1	0.8	4.0	4.8	4.8	8.8	4.8	5.6	5.6	4.9	4.4
+ QRT	4.0	0.0	1.6	0.8	**7.2**	1.6	4.0	**7.2**	4.0	7.2	3.8	0.8	4.8	3.2	6.4	3.2	4.0	5.6	3.2	3.9	3.8
+ M-SFT	3.2	0.0	4.8	0.8	3.2	4.8	4.8	6.4	4.0	8.0	4.0	1.6	4.8	3.2	4.0	4.0	4.0	4.0	3.2	3.6	3.8
+ Vanilla GRPO	4.0	**8.0**	4.0	**4.8**	4.0	6.4	**9.6**	5.6	5.6	8.0	**6.0**	6.4	4.8	5.6	4.0	5.6	6.4	3.2	7.2	5.4	5.7
+ LC-GRPO	3.2	3.2	8.0	0.8	4.8	6.4	8.0	6.4	**6.4**	8.0	5.5	2.4	5.6	5.6	**8.0**	6.4	6.4	**7.2**	**8.0**	6.2	5.8
+ M-Thinker	4.0	1.8	2.4	0.0	5.0	5.8	4.4	3.8	6.2	**8.2**	4.2	2.0	5.8	4.0	3.4	5.0	4.6	4.2	4.2	4.2	4.2
+ mGRPO	4.2	3.8	2.8	0.0	6.2	6.2	3.2	5.4	3.0	5.0	4.0	1.8	4.8	4.2	4.2	6.8	2.8	6.6	4.0	4.4	4.2
+ LANG	**4.8**	3.2	**8.8**	0.8	4.0	**10.4**	5.6	6.4	4.8	8.0	5.7	**6.4**	**6.4**	**6.4**	4.8	**8.8**	**7.2**	6.4	5.6	**6.2**	**5.9**
Qwen2.5-7B-Instruct	8.0	8.0	8.8	5.6	**12.0**	11.2	7.2	8.8	12.8	8.8	9.1	4.8	5.6	8.8	6.4	7.2	10.4	11.2	8.0	7.8	8.5
+ LCP	2.4	3.2	5.6	0.8	4.8	2.4	3.2	4.0	4.8	7.2	3.8	0.0	8.0	4.0	5.6	8.0	3.2	4.0	5.6	4.8	4.3
+ DIT	6.4	6.4	5.6	4.0	6.4	10.4	9.6	13.6	8.8	10.4	8.2	3.2	7.2	5.6	6.4	9.6	9.6	10.4	8.0	7.5	7.9
+ QRT	8.8	7.2	6.4	0.0	10.4	8.8	8.0	8.0	7.2	**13.6**	7.8	3.2	7.2	8.8	9.6	9.6	8.0	7.2	12.8	8.3	8.0
+ M-SFT	3.2	3.2	0.8	1.6	4.0	0.8	4.0	5.6	2.4	2.4	2.8	0.8	2.4	2.4	1.6	3.2	3.2	4.8	2.4	2.6	2.7
+ Vanilla GRPO	9.6	**10.4**	8.0	4.8	11.2	**11.2**	**12.0**	10.4	11.2	11.2	10.0	3.2	8.8	10.4	11.2	11.2	9.6	10.4	10.4	9.4	9.7
+ LC-GRPO	8.0	5.6	8.0	5.6	8.0	8.8	10.4	10.4	11.2	10.4	8.6	4.0	10.4	10.4	**12.8**	12.0	9.6	**12.8**	11.2	10.4	9.4
+ M-Thinker	8.0	5.2	5.4	3.4	5.4	6.6	6.6	10.4	6.6	9.0	6.7	5.4	6.0	7.4	8.0	8.4	9.2	9.2	10.2	8.0	7.2
+ mGRPO	10.4	8.8	8.0	**5.6**	11.2	9.6	10.4	12.0	10.4	8.8	9.5	**6.4**	8.8	10.4	9.6	13.6	**12.0**	9.6	12.0	10.3	9.9
+ LANG	**10.4**	6.4	**13.6**	3.2	10.4	10.4	8.8	**13.6**	**12.8**	11.2	**10.1**	4.0	**13.6**	**11.2**	9.6	**13.6**	8.0	10.4	**12.8**	**10.4**	**10.2**

Table 13: The accuracy (%) on PolyMath test sets of high-level difficulty.

Method	In-Domain Languages											Out-of-Domain Languages									ALL-Avg.
	Ar	Bn	Th	Sw	Ja	Zh	De	Fr	Ru	En	Avg.	Te	Ko	Ms	Vi	It	Id	Pt	Es	Avg.	
Qwen2.5-3B-Instruct	13.6	5.6	11.2	2.4	6.4	14.4	13.6	9.6	11.2	19.2	10.7	1.6	9.6	8.8	12.0	12.8	12.8	14.4	16.0	11.0	10.8
+ LCP	13.6	4.0	8.8	4.8	8.0	13.6	11.2	13.6	15.2	22.4	11.5	2.4	8.0	10.4	12.8	11.2	10.4	16.0	14.4	10.7	11.2
+ DIT	9.6	8.0	8.8	2.4	8.8	13.6	13.6	12.0	12.8	18.4	10.8	2.4	8.0	12.0	12.0	12.0	11.2	8.8	13.6	10.0	10.4
+ QRT	12.8	1.6	9.6	4.0	8.8	12.8	12.8	16.0	12.0	21.6	11.2	1.6	13.6	13.6	12.8	11.2	16.0	13.6	14.4	12.1	11.6
+ M-SFT	12.8	3.2	8.8	3.2	5.6	12.8	12.8	10.4	10.4	22.4	10.2	0.8	8.8	6.4	12.8	12.0	12.8	12.0	12.8	9.8	10.0
+ Vanilla GRPO	11.2	**16.8**	**13.6**	5.6	11.2	15.2	14.4	16.0	**18.4**	16.8	**13.9**	**9.6**	**16.0**	12.0	12.8	9.6	12.8	16.0	10.4	12.4	13.2
+ LC-GRPO	12.0	7.2	11.2	**6.4**	**15.2**	14.4	13.6	**21.6**	14.4	19.2	13.5	4.8	14.4	14.4	13.6	15.2	15.2	14.4	16.8	13.6	13.6
+ M-Thinker	**14.2**	6.2	11.8	3.8	10.2	**17.8**	13.0	9.8	13.4	19.6	12.0	4.0	10.4	15.6	15.8	13.8	11.2	13.4	12.6	12.1	12.0
+ mGRPO	10.2	6.2	10.2	3.0	13.8	15.8	**15.0**	15.2	14.2	**23.0**	12.7	4.8	12.6	9.8	13.4	14.8	9.4	10.8	13.8	11.2	12.0
+ LANG	10.4	7.2	12.8	4.8	13.6	17.6	14.4	17.6	17.6	20.8	13.7	8.0	12.0	**16.0**	**18.4**	**16.0**	**16.0**	**20.8**	**16.8**	**15.5**	**14.5**
Qwen2.5-7B-Instruct	14.4	16.0	19.2	8.8	16.0	24.8	23.2	24.8	24.8	32.8	20.5	12.8	20.0	20.8	22.4	22.4	22.4	21.6	23.2	20.7	20.6
+ LCP	12.0	7.2	10.4	2.4	8.0	12.8	14.4	13.6	12.8	20.8	11.4	1.6	12.0	12.8	12.8	12.8	13.6	12.8	18.4	12.1	11.7
+ DIT	15.2	14.4	20.8	8.8	18.4	20.0	20.8	30.4	24.0	29.6	20.2	10.4	17.6	16.0	17.6	22.4	20.0	23.2	24.0	18.9	19.6
+ QRT	15.2	8.0	16.8	8.0	20.8	24.0	22.4	26.4	24.8	26.4	19.3	11.2	24.8	20.0	21.6	20.0	20.0	23.2	20.0	20.1	19.6
+ M-SFT	6.4	5.6	14.4	4.0	8.0	11.2	9.6	8.8	10.4	10.4	8.9	0.8	9.6	4.8	9.6	10.4	4.0	13.6	8.0	7.6	8.3
+ Vanilla GRPO	24.0	**27.2**	**27.2**	18.4	22.4	28.0	28.0	28.0	27.2	**33.6**	26.4	10.4	**29.6**	24.8	24.0	28.0	24.8	28.8	23.2	24.2	25.4
+ LC-GRPO	20.0	16.0	16.8	20.8	27.2	24.8	30.4	30.4	24.8	32.8	24.4	12.8	25.6	22.4	24.8	24.0	28.0	25.6	30.4	24.2	24.3
+ M-Thinker	17.8	10.2	15.8	16.6	17.0	26.6	23.6	27.6	26.4	25.8	20.7	16.4	20.0	16.4	20.2	24.6	22.2	24.4	26.2	21.3	21.0
+ mGRPO	**31.2**	22.4	22.4	**21.6**	23.2	**31.2**	25.6	24.8	26.4	29.6	25.8	**20.0**	26.4	20.8	23.2	27.2	24.0	28.0	26.4	24.5	25.2
+ LANG	25.6	22.4	24.0	8.8	**29.6**	30.4	**33.6**	**33.6**	**29.6**	33.6	**27.1**	18.4	28.8	**28.8**	**29.6**	**36.0**	**31.2**	**29.6**	**31.2**	**29.2**	**28.0**

Table 14: The accuracy (%) on PolyMath test sets of medium-level difficulty.

Method	In-Domain Languages											Out-of-Domain Languages									ALL-Avg.
	Ar	Bn	Th	Sw	Ja	Zh	De	Fr	Ru	En	Avg.	Te	Ko	Ms	Vi	It	Id	Pt	Es	Avg.	
Qwen2.5-3B-Instruct	60.8	33.6	52.8	5.6	55.2	73.6	61.6	73.6	68.8	87.2	57.3	8.0	61.6	60.0	66.4	69.6	68.0	74.4	74.4	60.3	58.6
+ LCP	56.8	35.2	51.2	6.4	51.2	71.2	62.4	64.8	68.0	84.0	55.1	8.8	52.8	60.8	67.2	72.8	68.0	70.4	75.2	59.5	57.1
+ DIT	52.8	33.6	52.8	6.4	55.2	71.2	61.6	70.4	76.0	87.2	56.7	6.4	58.4	60.0	69.6	72.0	68.0	**78.4**	72.0	60.6	58.4
+ QRT	**68.8**	32.0	51.2	5.6	53.6	70.4	57.6	65.6	71.2	87.2	56.3	10.4	59.2	64.8	68.8	73.6	70.4	70.4	77.6	61.9	58.8
+ M-SFT	54.4	39.2	56.0	3.2	48.8	**75.2**	60.8	68.8	72.0	84.8	56.3	7.2	55.2	64.8	62.4	71.2	66.4	74.4	76.0	59.7	57.8
+ Vanilla GRPO	65.6	**47.2**	59.2	**11.2**	56.8	74.4	64.0	74.4	74.4	86.4	61.4	5.6	59.2	62.4	**72.0**	73.6	**73.6**	75.2	**79.2**	62.6	61.9
+ LC-GRPO	66.4	45.6	**65.6**	9.6	56.8	71.2	64.0	68.8	73.6	84.0	60.6	7.2	**64.0**	63.2	65.6	68.8	72.0	76.8	73.6	61.4	60.9
+ M-Thinker	55.6	36.4	54.6	4.8	57.4	74.0	61.2	69.0	74.4	85.6	57.3	8.8	61.2	63.4	66.6	74.8	68.6	71.8	74.4	61.2	59.0
+ mGRPO	59.4	44.4	57.8	6.8	58.0	71.8	65.6	73.6	72.2	85.0	59.5	10.2	58.4	59.2	71.2	73.8	68.2	72.8	72.2	60.8	60.0
+ LANG	61.6	45.6	62.4	8.8	**60.8**	74.4	**74.4**	**75.2**	**75.2**	**87.2**	**62.6**	**10.4**	62.4	**64.8**	69.6	**77.6**	71.2	72.0	76.8	**63.1**	**62.8**
Qwen2.5-7B-Instruct	81.6	64.8	75.2	16.0	68.8	77.6	77.6	82.4	84.0	88.8	71.7	28.8	74.4	78.4	83.2	83.2	84.0	82.4	86.4	75.1	73.2
+ LCP	58.4	36.8	56.0	4.8	52.0	70.4	60.0	70.4	74.4	84.8	56.8	11.2	60.0	63.2	64.0	68.8	67.2	74.4	76.0	60.6	58.5
+ DIT	80.8	62.4	75.2	16.0	72.0	76.8	78.4	77.6	83.2	91.2	71.4	33.6	71.2	76.8	79.2	85.6	79.2	79.2	84.0	74.2	72.6
+ QRT	**88.0**	63.2	78.4	18.4	70.4	80.0	76.0	79.2	84.0	92.0	73.0	35.2	73.6	76.8	77.6	80.8	82.4	80.8	86.4	74.2	73.5
+ M-SFT	35.2	38.4	55.2	4.8	33.6	30.4	26.4	37.6	35.2	61.6	35.8	10.4	57.6	34.4	65.6	72.8	39.2	59.2	38.4	47.2	40.9
+ Vanilla GRPO	84.8	69.6	76.0	15.2	69.6	**85.6**	81.6	83.2	85.6	91.2	74.2	31.2	78.4	80.8	81.6	83.2	80.8	86.4	**91.2**	76.7	75.3
+ LC-GRPO	79.2	68.0	76.8	19.2	74.4	83.2	78.4	82.4	87.2	92.8	74.2	36.0	76.0	80.8	83.2	83.2	84.0	85.6	86.4	76.9	75.4
+ M-Thinker	82.2	64.8	71.6	18.2	69.8	76.0	72.8	76.6	81.4	90.4	70.4	31.8	76.2	75.4	71.0	82.6	78.4	77.2	80.2	71.6	70.9
+ mGRPO	87.2	**76.0**	**80.8**	**36.0**	**78.4**	83.2	**82.4**	**86.4**	87.2	**94.4**	**79.2**	**44.8**	**80.0**	84.0	**85.6**	**88.8**	**85.6**	**88.0**	88.8	**80.7**	**79.9**
+ LANG	79.2	68.0	74.4	17.6	72.0	82.4	76.8	84.8	**88.0**	92.0	73.5	34.4	77.6	**84.0**	83.2	84.0	83.2	84.0	88.0	77.3	75.2

Table 15: The accuracy (%) on PolyMath test sets of low-level difficulty.

Method	In-Domain Languages											Out-of-Domain Languages									ALL-Avg.
	Ar	Bn	Th	Sw	Ja	Zh	De	Fr	Ru	En	Avg.	Te	Ko	Vi	It	Id	Pt	Es	Avg.		
Qwen2.5-3B-Instruct	3.2	2.4	2.4	0.8	5.6	3.2	3.2	4.8	2.4	1.6	3.0	0.0	2.4	4.8	4.0	4.0	**8.0**	5.6	4.1	3.4	
+ LCP	3.2	0.8	4.8	3.2	2.4	5.6	4.8	2.4	1.6	**6.4**	3.5	2.4	4.0	4.0	4.0	4.8	6.4	2.4	4.0	3.7	
+ DIT	3.2	4.0	2.4	**3.2**	1.6	4.8	**6.4**	4.8	4.8	4.0	3.9	0.0	4.0	2.4	1.6	4.0	4.8	4.0	3.0	3.5	
+ QRT	**6.4**	1.6	2.4	0.8	4.8	1.6	5.6	4.8	**6.4**	4.0	3.8	0.8	0.8	6.4	**6.4**	4.8	3.2	3.2	3.7	3.8	
+ M-SFT	1.6	2.4	1.6	0.8	1.6	3.2	0.8	1.6	4.0	4.0	2.2	0.0	1.6	0.8	1.6	0.0	1.6	1.6	1.0	1.7	
+ Vanilla GRPO	4.0	0.0	2.4	0.0	4.0	5.6	3.2	4.8	3.2	4.8	3.2	1.6	**4.8**	4.0	0.8	**7.2**	4.0	**5.6**	4.0	3.5	
+ LC-GRPO	5.6	3.2	4.8	2.4	4.8	6.4	3.2	4.8	3.2	4.0	4.2	2.4	1.6	4.0	4.8	6.4	4.8	3.2	3.9	4.1	
+ M-Thinker	4.8	1.6	1.6	2.4	3.2	3.2	5.6	2.4	4.0	4.8	3.4	0.8	3.2	4.0	4.0	1.6	4.8	3.2	3.1	3.2	
+ mGRPO	4.6	1.8	**5.2**	1.2	1.8	3.2	5.2	**6.6**	3.0	5.0	3.8	2.4	2.2	1.8	3.0	2.8	1.2	4.6	2.6	3.3	
+ LANG	4.0	**4.8**	4.8	2.4	**5.6**	**9.6**	4.8	4.0	5.6	4.8	**5.0**	**3.2**	4.0	**7.2**	4.0	3.2	5.6	4.0	**4.5**	**4.8**	
Qwen2.5-7B-Instruct	5.6	6.4	5.6	0.8	7.2	10.4	10.4	8.0	5.6	**8.8**	6.8	2.4	6.4	8.0	5.6	1.6	5.6	8.0	4.7	6.3	
+ LCP	5.6	1.6	3.2	0.8	4.8	1.6	4.0	3.2	5.6	7.2	3.8	0.0	3.2	3.2	3.2	1.6	5.6	3.2	2.9	3.4	
+ DIT	6.4	5.6	4.8	0.8	8.0	8.8	5.6	7.2	8.8	7.2	6.3	4.0	4.8	4.8	10.4	5.6	6.4	6.4	6.1	6.2	
+ QRT	4.0	**8.0**	5.6	**4.8**	**8.0**	8.0	8.8	**9.6**	8.0	8.0	7.3	**5.6**	4.0	6.4	5.6	6.4	8.8	8.8	6.5	7.0	
+ M-SFT	0.8	0.8	1.6	0.0	0.8	0.8	0.0	0.8	0.8	0.0	0.6	0.8	0.8	0.8	0.0	1.6	0.8	1.6	0.9	0.8	
+ Vanilla GRPO	4.8	0.0	0.0	0.0	5.6	**10.4**	8.0	7.2	6.4	7.2	5.0	0.0	5.6	8.0	6.4	5.6	8.0	8.0	5.9	5.4	
+ LC-GRPO	8.0	1.6	3.2	1.6	5.6	7.2	9.6	8.8	12.0	8.0	6.6	1.6	4.0	8.0	8.8	4.8	8.8	**11.2**	6.7	6.6	
+ M-Thinker	6.8	3.2	4.4	0.0	6.4	5.6	7.2	7.0	7.4	6.0	5.4	3.8	4.0	5.2	7.2	5.8	6.4	7.8	5.7	5.5	
+ mGRPO	7.2	3.2	3.2	0.0	4.0	6.4	6.4	6.4	8.8	5.6	5.1	3.2	4.0	5.6	5.6	6.4	8.0	8.0	5.8	5.4	
+ LANG	**8.0**	4.0	**6.4**	1.6	7.2	9.6	**11.2**	8.0	**12.0**	8.0	**7.6**	3.2	**6.4**	**9.6**	**12.0**	**10.4**	**9.6**	10.4	**8.8**	**8.1**	

Table 16: The LC&Acc (%) on PolyMath test sets of top-level difficulty.

Method	In-Domain Languages											Out-of-Domain Languages								ALL-Avg.
	Ar	Bn	Th	Sw	Ja	Zh	De	Fr	Ru	En	Avg.	Te	Ko	Vi	It	Id	Pt	Es	Avg.	
Qwen2.5-3B-Instruct	3.2	0.8	4.8	0.8	4.0	3.2	6.4	3.2	5.6	8.0	4.0	0.8	6.4	4.8	4.0	4.8	2.4	4.8	4.0	4.0
+ LCP	3.2	1.6	4.8	2.4	4.0	5.6	6.4	7.2	5.6	8.0	4.9	0.8	2.4	3.2	4.0	4.8	3.2	6.4	3.5	4.3
+ DIT	4.0	0.0	1.6	**2.4**	6.4	4.8	8.8	4.0	4.0	4.8	4.1	0.8	4.0	4.8	8.8	4.0	5.6	5.6	4.8	4.4
+ QRT	4.0	0.0	1.6	0.8	**7.2**	1.6	4.0	**7.2**	4.0	7.2	3.8	0.8	4.8	6.4	3.2	4.0	5.6	3.2	4.0	3.9
+ M-SFT	0.8	0.8	3.2	0.8	1.6	3.2	1.6	1.6	3.2	1.6	1.8	0.8	2.4	2.4	1.6	0.8	1.6	3.2	1.8	1.8
+ Vanilla GRPO	4.0	0.0	4.0	0.8	4.0	6.4	**9.6**	5.6	5.6	8.0	4.8	4.0	1.6	4.0	4.8	6.4	3.2	7.2	4.5	4.7
+ LC-GRPO	3.2	3.2	8.0	0.8	4.8	6.4	8.0	6.4	6.4	8.0	**5.5**	2.4	5.6	**8.0**	6.4	6.4	**7.2**	**8.0**	**6.3**	**5.8**
+ M-Thinker	4.8	2.4	2.4	0.8	4.0	3.2	4.0	4.0	**7.2**	4.8	3.8	0.0	4.8	2.4	3.2	1.6	3.2	3.2	2.6	3.3
+ mGRPO	4.2	**3.8**	2.8	0.0	6.2	4.2	3.2	5.4	3.0	5.0	3.8	1.8	4.8	4.2	6.8	2.8	5.6	4.0	4.3	4.0
+ LANG	**4.8**	3.2	**8.8**	0.8	4.0	**8.0**	5.6	6.4	4.8	**8.0**	5.4	**4.0**	**6.4**	4.8	**8.8**	**7.2**	6.4	5.6	6.2	5.7
Qwen2.5-7B-Instruct	8.0	7.2	8.0	1.6	**12.0**	8.8	6.4	8.0	11.2	8.8	8.0	4.8	5.6	6.4	7.2	4.8	9.6	8.0	6.6	7.4
+ LCP	2.4	3.2	5.6	0.8	4.8	2.4	3.2	4.0	4.8	7.2	3.8	0.0	8.0	5.6	8.0	3.2	4.0	5.6	4.9	4.3
+ DIT	6.4	6.4	5.6	1.6	6.4	9.6	9.6	13.6	8.8	10.4	7.8	3.2	7.2	6.4	9.6	9.6	10.4	8.0	7.8	7.8
+ QRT	8.8	**7.2**	6.4	0.0	10.4	7.2	8.0	8.0	7.2	**13.6**	7.7	3.2	4.8	9.6	9.6	8.0	7.2	12.8	7.9	7.8
+ M-SFT	2.4	0.0	0.8	1.6	4.0	0.8	4.0	4.0	3.2	2.4	2.3	0.8	0.0	1.6	3.2	0.0	4.8	2.4	1.8	2.1
+ Vanilla GRPO	9.6	0.0	0.0	0.8	9.6	8.0	**12.0**	10.4	11.2	11.2	7.3	3.2	8.0	11.2	11.2	9.6	10.4	10.4	9.1	8.0
+ LC-GRPO	8.0	5.6	8.0	0.0	8.0	8.0	10.4	10.4	11.2	10.4	8.0	4.0	10.4	**12.8**	12.0	**9.6**	**12.8**	11.2	**10.4**	9.0
+ M-Thinker	8.0	5.2	5.4	3.2	5.4	5.6	6.6	10.4	6.6	8.0	6.4	**5.4**	5.6	8.0	7.2	8.0	8.0	10.2	7.5	6.9
+ mGRPO	9.6	5.6	8.0	1.6	10.4	8.8	7.2	12.0	10.4	8.8	8.2	3.2	8.8	8.0	13.6	4.0	9.6	12.0	8.5	8.3
+ LANG	**10.4**	6.4	**13.6**	**3.2**	10.4	**9.6**	8.8	**13.6**	**12.8**	11.2	**10.0**	4.0	**13.6**	9.6	**13.6**	8.0	10.4	**12.8**	10.3	**10.1**

Table 17: The LC&Acc (%) on PolyMath test sets of high-level difficulty.

Method	In-Domain Languages											Out-of-Domain Languages								ALL-Avg.
	Ar	Bn	Th	Sw	Ja	Zh	De	Fr	Ru	En	Avg.	Te	Ko	Vi	It	Id	Pt	Es	Avg.	
Qwen2.5-3B-Instruct	12.8	5.6	10.4	2.4	5.6	12.8	13.6	9.6	11.2	19.2	10.3	1.6	9.6	12.0	12.8	12.8	14.4	15.2	11.2	10.7
+ LCP	**13.6**	3.2	8.8	4.8	8.0	13.6	11.2	13.6	15.2	22.4	11.4	2.4	8.0	12.8	11.2	10.4	16.0	14.4	10.7	11.2
+ DIT	9.6	7.2	8.8	2.4	8.8	12.0	13.6	12.0	12.8	18.4	10.6	2.4	8.0	12.0	12.0	11.2	8.8	13.6	9.7	10.2
+ QRT	12.8	1.6	9.6	4.0	8.8	12.8	12.8	16.0	12.0	21.6	11.2	1.6	13.6	12.8	11.2	16.0	13.6	14.4	11.9	11.5
+ M-SFT	0.8	3.2	3.2	1.6	4.8	4.0	4.8	2.4	6.4	3.2	3.4	0.8	1.6	6.4	2.4	1.6	4.0	6.4	3.3	3.4
+ Vanilla GRPO	11.2	0.8	**13.6**	4.8	11.2	14.4	13.6	16.0	**18.4**	16.8	12.1	2.4	12.0	12.8	9.6	12.8	16.0	10.4	10.9	11.6
+ LC-GRPO	12.0	7.2	10.4	**6.4**	**15.2**	14.4	13.6	**21.6**	14.4	19.2	13.4	4.8	**14.4**	13.6	15.2	15.2	14.4	16.0	13.4	13.4
+ M-Thinker	11.2	6.4	7.2	3.2	10.4	13.6	11.2	10.4	14.4	21.6	11.0	2.4	9.6	12.8	13.6	13.6	13.6	12.8	11.2	11.1
+ mGRPO	9.6	6.2	10.2	1.6	8.0	12.8	**15.0**	15.2	13.6	**23.0**	11.5	4.8	11.2	13.4	13.6	9.4	10.8	13.8	11.0	11.3
+ LANG	10.4	**7.2**	12.8	4.8	13.6	**16.0**	14.4	17.6	17.6	20.8	**13.5**	**8.0**	12.0	**18.4**	**16.0**	**16.0**	**20.8**	**16.8**	**15.4**	**14.3**
Qwen2.5-7B-Instruct	12.8	16.0	19.2	2.4	16.0	23.2	21.6	24.8	24.8	32.8	19.4	12.8	20.0	21.6	22.4	14.4	20.0	22.4	19.1	19.3
+ LCP	11.2	7.2	10.4	2.4	6.4	12.0	14.4	13.6	12.8	20.8	11.1	1.6	12.0	12.8	12.8	13.6	12.8	18.4	12.0	11.5
+ DIT	13.6	14.4	20.8	6.4	18.4	17.6	20.8	30.4	23.2	29.6	19.5	10.4	17.6	17.6	22.4	20.0	23.2	24.0	19.3	19.4
+ QRT	14.4	8.0	16.0	8.0	20.8	22.4	22.4	26.4	24.8	26.4	19.0	11.2	24.0	21.6	20.0	20.0	23.2	20.0	20.0	19.4
+ M-SFT	10.4	4.0	14.4	4.0	8.0	8.0	9.6	6.4	12.8	10.4	8.8	0.8	5.6	9.6	10.4	4.0	13.6	8.8	7.5	8.3
+ Vanilla GRPO	22.4	3.2	6.4	6.4	21.6	26.4	28.0	28.0	24.0	33.6	20.0	9.6	22.4	23.2	28.0	24.0	27.2	23.2	22.5	21.0
+ LC-GRPO	20.0	16.0	16.8	4.8	27.2	22.4	30.4	30.4	24.8	32.8	22.6	12.8	24.8	24.8	24.0	28.0	25.6	30.4	24.3	23.3
+ M-Thinker	17.8	10.2	3.6	8.0	16.4	24.0	20.8	24.0	26.4	22.4	17.4	16.4	16.0	20.2	22.4	22.2	24.4	24.8	20.9	18.8
+ mGRPO	22.4	22.4	22.4	**9.6**	23.2	24.0	25.6	24.8	26.4	29.6	23.0	18.4	26.4	23.2	27.2	24.0	28.0	26.4	24.8	23.8
+ LANG	**25.6**	**22.4**	**24.0**	8.8	**29.6**	**28.0**	**33.6**	**33.6**	**29.6**	**33.6**	**26.9**	**18.4**	**28.8**	**29.6**	**36.0**	**31.2**	**29.6**	**31.2**	**29.3**	**27.9**

Table 18: The LC&Acc (%) on PolyMath test sets of medium-level difficulty.

Method	In-Domain Languages											Out-of-Domain Languages								ALL-Avg.
	Ar	Bn	Th	Sw	Ja	Zh	De	Fr	Ru	En	Avg.	Te	Ko	Vi	It	Id	Pt	Es	Avg.	
Qwen2.5-3B-Instruct	60.0	33.6	52.8	5.6	55.2	55.2	60.0	73.6	67.2	87.2	55.0	8.0	61.6	66.4	68.8	67.2	68.0	73.6	59.1	56.7
+ LCP	56.8	35.2	51.2	6.4	51.2	58.4	62.4	63.2	66.4	84.0	53.5	8.8	52.8	67.2	72.8	68.0	68.0	75.2	59.0	55.8
+ DIT	52.8	33.6	52.8	6.4	54.4	56.0	61.6	69.6	75.2	87.2	55.0	6.4	58.4	69.6	71.2	68.0	74.4	71.2	59.9	57.0
+ QRT	**68.8**	32.0	50.4	5.6	53.6	58.4	56.0	64.8	71.2	87.2	54.8	10.4	59.2	68.8	72.8	70.4	66.4	**77.6**	60.8	57.3
+ M-SFT	23.2	12.8	32.8	0.8	12.0	22.4	22.4	19.2	30.4	22.4	19.8	0.8	48.8	52.0	22.4	27.2	28.0	46.4	32.2	24.9
+ Vanilla GRPO	65.6	4.8	59.2	**10.4**	56.8	61.6	62.4	74.4	73.6	86.4	55.5	5.6	59.2	**72.0**	72.8	**73.6**	72.8	79.2	62.2	58.3
+ LC-GRPO	66.4	45.6	**65.6**	9.6	56.8	56.8	64.0	68.0	72.8	84.0	59.0	7.2	**64.0**	65.6	68.0	72.0	**76.0**	73.6	60.9	59.8
+ M-Thinker	56.0	36.8	60.0	6.4	47.2	**65.6**	61.6	68.0	70.4	87.2	55.9	5.6	56.8	65.6	68.8	66.4	68.0	74.4	57.9	56.8
+ mGRPO	58.4	44.4	57.8	1.6	58.0	56.0	65.6	71.2	72.2	85.0	57.0	10.2	58.4	71.2	73.8	68.2	72.8	71.2	60.8	58.6
+ LANG	61.6	**45.6**	62.4	8.8	**60.8**	61.6	**74.4**	**74.4**	**74.4**	**87.2**	**61.1**	**10.4**	62.4	69.6	**77.6**	71.2	68.8	76.0	**62.3**	**61.6**
Qwen2.5-7B-Instruct	78.4	64.8	75.2	15.2	68.8	64.8	77.6	82.4	82.4	88.8	69.8	28.8	72.0	83.2	83.2	84.0	80.8	86.4	74.1	71.6
+ LCP	57.6	36.8	56.0	4.8	52.0	55.2	59.2	70.4	73.6	84.8	55.0	11.2	60.0	64.0	67.2	66.4	71.2	76.0	59.4	56.8
+ DIT	77.6	62.4	74.4	16.0	72.0	61.6	78.4	76.0	83.2	91.2	69.3	33.6	71.2	79.2	85.6	79.2	76.8	84.0	72.8	70.7
+ QRT	86.4	63.2	**78.4**	**18.4**	70.4	64.0	75.2	79.2	83.2	92.0	71.0	35.2	72.8	77.6	80.8	81.6	78.4	86.4	73.3	72.0
+ M-SFT	40.0	27.2	55.2	4.8	33.6	29.6	26.4	41.6	35.2	61.6	35.5	10.4	48.8	65.6	72.8	29.6	59.2	48.8	47.9	40.6
+ Vanilla GRPO	75.2	61.6	8.8	15.2	69.6	**68.8**	**81.6**	**83.2**	**85.6**	91.2	64.1	31.2	64.0	81.6	82.4	80.0	85.6	**91.2**	73.7	68.0
+ LC-GRPO	79.2	**68.0**	76.8	14.4	**74.4**	66.4	78.4	81.6	84.8	92.8	**71.7**	36.0	76.0	83.2	83.2	84.0	84.0	84.8	**75.9**	**73.4**
+ M-Thinker	82.2	64.8	71.6	18.2	69.8	66.4	72.8	76.6	81.4	90.4	69.4	31.8	76.2	71.0	82.6	78.4	74.4	80.2	70.7	69.9
+ mGRPO	82.4	66.4	70.4	14.6	70.4	56.0	65.6	71.2	72.2	85.0	57.0	10.2	58.4	71.2	73.8	**84.0**	**88.0**	71.2	60.8	58.6
+ LANG	**86.4**	45.6	62.4	8.8	60.8	61.6	74.4	74.4	74.4	**92.8**	61.1	**36.0**	**78.4**	**84.0**	**85.6**	71.2	68.8	76.0	62.3	61.6

Table 19: The LC&Acc (%) on PolyMath test sets of low-level difficulty.

Method	XWinograd						
	pt	ru	fr	jp	zh	en	Avg.
Qwen2.5-7B-Instruct	68.8	61.6	71.1	61.5	61.3	68.4	65.4
+ Vanilla GRPO	64.6	57.7	66.3	61.7	57.1	64.2	61.9
+ LC-GRPO	63.9	59.1	68.7	60.8	54.6	66.5	62.3
+ LANG	**79.1**	**70.8**	**72.3**	**72.1**	**81.0**	**84.5**	**76.6**

Table 20: The accuracy (%) for all languages on the XWinograd test sets.

Method	XStoryCloze											
	eu	my	sw	te	hi	id	ar	es	ru	zh	en	Avg.
Qwen2.5-7B-Instruct	49.8	49.6	53.0	55.9	58.8	64.3	63.6	67.2	68.5	66.2	72.5	60.9
+ Vanilla GRPO	50.2	49.2	51.0	56.3	56.6	58.1	60.6	61.2	63.4	60.4	65.9	57.5
+ LC-GRPO	**52.1**	49.4	**52.5**	**56.3**	58.6	63.7	62.2	64.7	67.6	64.7	70.2	60.2
+ LANG	51.0	**50.7**	52.3	56.2	**61.5**	**68.2**	**64.0**	**70.9**	**71.5**	**71.5**	**80.9**	**63.5**

Table 21: The accuracy (%) for all languages on the XStoryCloze test sets.

Method	XCOPA											
	et	ht	qu	sw	ta	id	th	tr	vi	it	zh	Avg.
Qwen2.5-7B-Instruct	**53.6**	50.2	50.6	**54.0**	54.2	69.2	62.6	61.4	68.8	72.2	71.2	60.7
+ Vanilla GRPO	51.2	49.6	49.6	52.6	**56.0**	66.0	61.8	59.2	65.4	67.2	69.2	58.9
+ LC-GRPO	51.4	51.4	**50.8**	50.6	55.6	69.2	**63.2**	61.4	69.0	71.0	71.4	60.5
+ LANG	50.6	**53.0**	49.4	52.6	55.4	**74.0**	61.0	**62.0**	**78.2**	**75.6**	**80.2**	**62.9**

Table 22: The accuracy (%) for all languages on the XCOPA test sets.

Method	MMLU-ProX					
	zu	yo	wo	ur	te	ne
Qwen2.5-7B-Instruct	8.7	6.4	16.5	10.9	25.7	19.5
+ Vanilla GRPO	0.1	0.0	0.1	1.1	0.2	0.0
+ LC-GRPO	10.4	19.2	10.9	30.9	22.4	28.5
+ LANG	**15.7**	**20.9**	**21.3**	**34.5**	**26.4**	**31.4**

Method	MMLU-ProX					
	mr	sw	bn	af	sr	uk
Qwen2.5-7B-Instruct	16.1	**23.6**	26.2	41.2	41.2	40.8
+ Vanilla GRPO	0.1	5.4	0.9	11.8	39.7	26.5
+ LC-GRPO	30.1	19.5	26.4	44.2	**44.6**	45.2
+ LANG	**32.0**	6.6	**32.2**	**45.7**	44.4	**46.0**

Method	MMLU-ProX					
	hu	hi	vi	id	ar	th
Qwen2.5-7B-Instruct	35.7	32.0	47.8	48.6	36.7	39.4
+ Vanilla GRPO	10.4	1.4	46.9	**51.0**	4.5	6.4
+ LC-GRPO	34.1	35.6	**50.1**	47.3	**45.0**	**43.9**
+ LANG	**39.4**	**36.0**	50.0	50.2	44.3	43.6

Method	MMLU-ProX					
	ko	cs	ru	it	pt	ja
Qwen2.5-7B-Instruct	34.9	44.1	40.1	49.5	50.3	46.3
+ Vanilla GRPO	27.4	45.3	6.6	48.0	44.7	3.9
+ LC-GRPO	45.4	**46.8**	**51.0**	**53.0**	52.6	48.1
+ LANG	**45.6**	46.5	50.7	52.5	**53.1**	**48.7**

Method	MMLU-ProX					
	de	fr	es	zh	en	Avg.
Qwen2.5-7B-Instruct	48.3	50.7	51.1	51.2	56.2	35.9
+ Vanilla GRPO	36.3	43.5	36.7	41.5	33.6	19.8
+ LC-GRPO	**51.1**	52.4	52.7	48.9	56.9	39.6
+ LANG	51.0	**52.8**	**53.3**	**53.6**	**59.4**	**41.0**

Table 23: The extract match (EM) (%) for all languages on the MMLU-ProX test sets.

Lang.	Native Prompt	LCP Prompt	DIT Prompt	QRT Prompt
En	{input}\nPlease reason step by step, and put your final answer within \\boxed{{}}.	{input}\nLet's think step by step and output the final answer within \\boxed{{}}. Use English to think and answer.	{input}\nLet's think step by step and output the final answer within \\boxed{{}}. Alright, Okay	OK, so the problem is {input}. Let me think in English.\nAnd put the final answer inside \\boxed{{}}. First
Es	{input}\nPor favor, razona paso a paso y pon tu respuesta final dentro de \\boxed{{}}.	{input}\nPensemos paso a paso y escribamos la respuesta final dentro de \\boxed{{}}. Usa español para pensar y responder.	{input}\nPensemos paso a paso y escribamos la respuesta final dentro de \\boxed{{}}. Buneo	Bien, el problema es {input}. Déjame pensar en español.\nY pon la respuesta final dentro de \\boxed{{}}. Primero
Fr	{input}\nVeuillez raisonner étape par étape et mettre votre réponse finale dans \\boxed{{}}.	{input}\nRéfléchissons pas à pas et inscrivons la réponse finale dans \\boxed{{}}. Utilisez le français pour penser et répondre.	{input}\nRéfléchissons pas à pas et inscrivons la réponse finale dans \\boxed{{}}. Bon	D'accord, donc le problème est {input}. Laissez-moi réfléchir en français.\nEt mettez la réponse finale dans \\boxed{{}}. D'abord
Zh	{input}\n请逐步推理，并将您的最终答案放在 \\boxed{{}} 中。	{input}\n让我们一步一步地思考，并输出最终答案在\\boxed{{}}中。使用中文进行思考和回答。	{input}\n让我们一步一步地思考，并输出最终答案在\\boxed{{}}中。嗯，好	好的，问题是{input}。让我用中文思考一下。\n并输出最终答案在\\boxed{{}}中。首先
Ja	{input}\nステップバイステップで推論し、最終的な答えを \\boxed{{}} の中に入れてください。	{input}\n一歩ずつ考え、最終的な答えを\\boxed{{}}に出力しましょう。日本語を使って考え、回答してください。	{input}\n一歩ずつ考え、最終的な答えを\\boxed{{}}に出力しましょう。まず	わかりました。問題は{input}です。日本語で考えさせてください。\n最終的な答えを \\boxed{{}} の中に入れます。まず
Th	{input}\nกรุณาเหตุผลขั้นตอนต่อขั้นตอนและใส่คำตอบสุดท้ายของคุณใน \\boxed{{}}.	{input}\nให้เราคิดทีละขั้นตอนและแสดงคำตอบสุดท้ายไว้ใน \\boxed{{}}. ใช้ภาษาไทยในการคิดและตอบคำถาม.	{input}\nให้เราคิดทีละขั้นตอนและแสดงคำตอบสุดท้ายไว้ใน \\boxed{{}}. โอเค	ตกลง ปัญหาคือ {input}. ขอให้ฉันได้คิดเป็นภาษาไทยก่อนนะ\nและใส่คำตอบสุดท้ายไว้ใน \\boxed{{}} ก่อนอื่น
Ko	{input}\n단계별로 추론하고 최종 답변을 \\boxed{{}} 안에 넣어주세요.	{input}\n단계적으로 생각하고 최종 답을 \\boxed{{}} 안에 출력합시다. 한국어로 생각하고 답변하세요	{input}\n단계적으로 생각하고 최종 답을 \\boxed{{}} 안에 출력합시다. 좋아	좋습니다. 문제는 {input}입니다. 한국어로 생각해 보겠습니다.\n최종 답을 \\boxed{{}} 안에 넣겠습니다. 먼저
Pt	{input}\nPor favor, raciocine passo a passo e coloque sua resposta final dentro de \\boxed{{}}.	{input}\nVamos pensar passo a passo e apresentar a resposta final dentro de \\boxed{{}}. Use português para pensar e responder.	{input}\nVamos pensar passo a passo e apresentar a resposta final dentro de \\boxed{{}}. Ok, Bem	Ok, então o problema é {input}. Deixe-me pensar em português.\nE coloque a resposta final dentro de \\boxed{{}}. Primeiro
Vi	{input}\nVui lòng lý giải từng bước và đặt câu trả lời cuối cùng của bạn trong \\boxed{{}}.	{input}\nHãy suy nghĩ từng bước một và đưa ra câu trả lời cuối cùng trong \\boxed{{}}. Sử dụng tiếng Việt để suy nghĩ và trả lời.	{input}\nHãy suy nghĩ từng bước một và đưa ra câu trả lời cuối cùng trong \\boxed{{}}. Được rồi, Đầu tiên	Được rồi, vấn đề là {input}. Hãy để tôi nghĩ bằng tiếng Việt.\nVà đặt câu trả lời cuối cùng vào trong \\boxed{{}}. Đầu tiên
Ar	{input}\n يرجى المنطق خطوة بخطوة، ووضع إجابتك النهائية داخل \\boxed{{}}.	{input}\n دعونا نفكر خطوةً بخطوة، ونضع الجواب النهائي داخل \\boxed{{}}. استخدم العربية للتفكير و الإجابة.	{input}\n دعونا نفكر خطوةً بخطوة، ونضع الجواب النهائي داخل \\boxed{{}}. حسنا	{input}. حسنًا، المشكلة هي \n دعني أفكر باللغة العربية. ونضع الجواب النهائي داخل \\boxed{{}}.

Figure 9: Prompts utilized to evaluate different methods on the MMATH test sets.

Lang.	Native Prompt	LCP Prompt	DIT Prompt	QRT Prompt
En	{input}\nNote: Please put the final answer in the \\boxed{{}}.	{input}\nNote: Please put the final answer in the \\boxed{{}}. Use English to think and answer.	{input}\nLet's think step by step and output the final answer within \\boxed{{}}. Alright, Okay.	OK, so the problem is {input}. Let me think in English.\nAnd put the final answer inside \\boxed{{}}.
Ar	{input}\n ملاحظة: يُرجى وضع الإجابة النهائية في \\boxed{{}}.	{input}\n ملاحظة: يُرجى وضع الإجابة النهائية في \\boxed{{}}. استخدم العربية للتفكير والإجابة.	{input}\n دعونا نفكر خطوةً بخطوة، ونضع الجواب النهائي داخل \\boxed{{}}. حسنا.	{input}\n حسنًا، المشكلة هي {input}. دعني أفكر باللغة العربية . ونضع الجواب النهائي داخل \n \\boxed{{}}.
Bn	{input}\nবিঃদ্রঃ অনুগ্রহ করে চূড়ান্ত উত্তরটি \\boxed{{}} এর মধ্যে রাখুন। .	{{input}\n বিঃদ্রঃ অনুগ্রহ করে চূড়ান্ত উত্তরটি \\boxed{{}} এর মধ্যে রাখুন। . বাংলা বযবহার করর চিন্তা এবং উত্তর চি ন।	{input}\n চলুন আমরা ধাপে ধাপে চিন্তা করি এবং চূড়ান্ত উত্তরটি \\boxed{{}} আচ্ছা, ঠিক আছে.	ঠিক আছে, প্রশ্নটি হলো {input}। আমাকে বাংলায় চিন্তা করতে দিন। \nএবং চূড়ান্ত উত্তরটি \\boxed{{}} এর মধ্যে রাখব। প্রথমে
Th	{input}\nหมายเหตุ: กรุณาใส่คำตอบสุดท้ายใน \\boxed{{}}.	{input}\n หมายเหตุ: กรุณาใส่คำตอบสุดท้ายใน \\boxed{{}}. ใช้ภาษาไทยในการคิดและตอบคำถาม.	{input}\n ให้เราคิดทีละขั้นตอนและแสดงคำตอบสุดท้ายไว้ใน \\boxed{{}}. โอเค.	ตกลง ปัญหาคือ {input}. ขอให้ฉันได้คิดเป็นภาษาไทยก่อนนะ\n และใส่คำตอบสุดท้ายไว้ใน \\boxed{{}} ก่อนอื่น
Sw	{input}\nKumbuka: Tafadhali weka jibu la mwisho katika \\boxed{{}}.	{input}\nKumbuka: Tafadhali weka jibu la mwisho katika \\boxed{{}}. Tumia Kiswahili kufikiri na kujibu.	{input}\nTufikirie hatua kwa hatua na tutoe jibu la mwisho ndani ya \\boxed{{}}. Sawa, vizuri.	{input}\nSawa, tatizo ni kwamba {input}. Wacha nifikirie kwa Kiswahili.\nNa tuweke jibu la mwisho ndani ya \\boxed{{}}. Kwanza
De	{input}\nHinweis: Bitte setzen Sie die endgültige Antwort in \\boxed{{}}.	{input}\nHinweis: Bitte setzen Sie die endgültige Antwort in \\boxed{{}}.	{input}\nDenken wir Schritt für Schritt und geben die endgültige Antwort in \\boxed{{}} an. Also, gut.	Gut, die Frage lautet {input}. Lass mich auf Deutsch nachdenken.\nUnd wir werden die endgültige Antwort in \\boxed{{}} setzen. Zuerst
Fr	{input}\nRemarque : Veuillez mettre la réponse finale dans le \\boxed{{}}.	{input}\nRemarque : Veuillez mettre la réponse finale dans le \\boxed{{}}. Utilisez le français pour penser et répondre.	{input}\nRéfléchissons pas à pas et inscrivons la réponse finale dans \\boxed{{}}. Bon.	D'accord, donc le problème est {input}. Laissez-moi réfléchir en français.\nEt mettez la réponse finale dans \\boxed{{}}. D'abord
Ja	{input}\n注意：最終的な答えを \\boxed{{}} に入れてください。	{input}\n注意：最終的な答えを \\boxed{{}} に入れてください。日本語を使って考え、回答してください。	{input}\n一歩ずつ考え、最終的な答えを\\boxed{{}}に出力しましょう。まず。	わかりました。問題は{input}です。日本語で考えさせてください。\n最終的な答えを \\boxed{{}} の中に入れます。
Ko	{input}\n참고: 최종 답안을 \\boxed{{}} 안에 넣어 주세요.	{input}\n참고: 최종 답안을 \\boxed{{}} 안에 넣어 주세요. 한국어로 생각하고 답변하세요.	{input}\n단계적으로 생각하고 최종 답을 \\boxed{{}} 안에 출력합시다. 좋아.	좋습니다. 문제는 {input}입니다. 한국어로 생각해 보겠습니다.\n최종 답을 \\boxed{{}} 안에 넣겠습니다. 먼저
Ms	{input}\nNota: Sila letakkan jawapan akhir dalam \\boxed{{}}.	{input}\nNota: Sila letakkan jawapan akhir dalam \\boxed{{}}. Gunakan bahasa Melayu untuk berfikir dan menjawab.	{input}\nMari kita fikirkan langkah demi langkah dan keluarkan jawapan akhir dalam \\boxed{{}}. Baiklah, ok.	Baik, masalahnya ialah {input}. Biar saya berfikir dalam bahasa Melayu.\nDan letakkan jawapan akhir di dalam \\boxed{{}}. Pertama
Vi	user\n{input}\nLưu ý: Vui lòng đặt câu trả lời cuối cùng trong \\boxed{{}}.	{input}\nLưu ý: Vui lòng đặt câu trả lời cuối cùng trong \\boxed{{}}. Sử dụng tiếng Việt để suy nghĩ và trả lời.	{input}\nHãy suy nghĩ từng bước một và đưa ra câu trả lời cuối cùng trong \\boxed{{}}. Được rồi, Đầu tiên.	Được rồi, vấn đề là {input}. Hãy để tôi nghĩ bằng tiếng Việt.\nVà đặt câu trả lời cuối cùng vào trong \\boxed{{}}. Đầu tiên
Zh	{input}\n注意：请将最终答案放在 \\boxed{{}} 中。	{input}\n注意：请将最终答案放在 \\boxed{{}} 中。使用中文进行思考和回答。	{input}\n让我们一步一步地思考，并输出最终答案在 \\boxed{{}}中。嗯，好。	好的，问题是{input}。让我用中文思考一下。\n并输出最终答案在\\boxed{{}}中。首先
Ru	{input}\nПримечание: Пожалуйста, поместите окончательный ответ в \\boxed{{}}.	{input}\nПримечание: Пожалуйста, поместите окончательный ответ в \\boxed{{}}. Используйте русский язык для размышлений и ответов.	{input}\nДавайте подумаем шаг за шагом и выведем окончательный ответ внутри \\boxed{{}}. Хорошо, ладно.	{input}\nХорошо, задача заключается в том, что {input}. Позвольте мне подумать по-русски.\nИ поместим окончательный ответ в \\boxed{{}}. Сначала
Es	{input}\nNota: Por favor, coloque la respuesta final en el \\boxed{{}}.	{input}\nNota: Por favor, coloque la respuesta final en el \\boxed{{}}. Usa español para pensar y responder.	{input}\nPensemos paso a paso y escribamos la respuesta final dentro de \\boxed{{}}. Buneo.	Bien, el problema es {input}. Déjame pensar en español.\nY pon la respuesta final dentro de \\boxed{{}}. Primero
Te	{input}\nగమనిక: దయచేసి తుది జవాబును \\boxed{{}} లో ఉంచండి.	{input}\nగమనిక: దయచేసి తుది జవాబును \\boxed{{}} లో ఉంచండి. తెలుగును ఉపయోగించి ఆలోచించండి సమాధానం ఇవ్వండి .	{input}\nప్రతి దశను పరిగణనలోకి తీసుకుని \\boxed{{}} లో ఫైనల్ సమాధానాన్ని ఇవ్వండి. సరే, బాగుంది.	{input}\nసరే, సమస్య {input}. నేను తెలుగు లో ఆలోచించనివ్వండి.\nమరియు తుది సమాధానాన్ని \\boxed{{}} లో ఉంచండి. మొదట
It	{input}\nNota: Per favore, metti la risposta finale nel \\boxed{{}}.	{input}\nNota: Per favore, metti la risposta finale nel \\boxed{{}}. Usa italiano per pensare e rispondere.	{input}\nRagioniamo passo dopo passo e inseriamo la risposta finale all'interno di \\boxed{{}}. Bene, ok.	D'accordo, quindi il problema è {input}. Lasciami pensare in italiano.\nE metti la risposta finale dentro \\boxed{{}}. Per prima cosa
Id	{input}\nCatatan: Silakan letakkan jawaban akhir di dalam \\boxed{{}}.	{input}\nCatatan: Silakan letakkan jawaban akhir di dalam \\boxed{{}}. Gunakan bahasa Indonesia untuk berpikir dan menjawab.	{input}\nMari kita berpikir langkah demi langkah dan keluarkan jawaban akhir di dalam \\boxed{{}}. Baiklah, oke.	Baik, masalahnya adalah {input}. Biarkan saya berpikir dalam bahasa Indonesia.\nDan letakkan jawaban akhirnya di dalam \\boxed{{}}. Pertama

Figure 10: Prompts utilized to evaluate different methods on the PolyMath test sets.

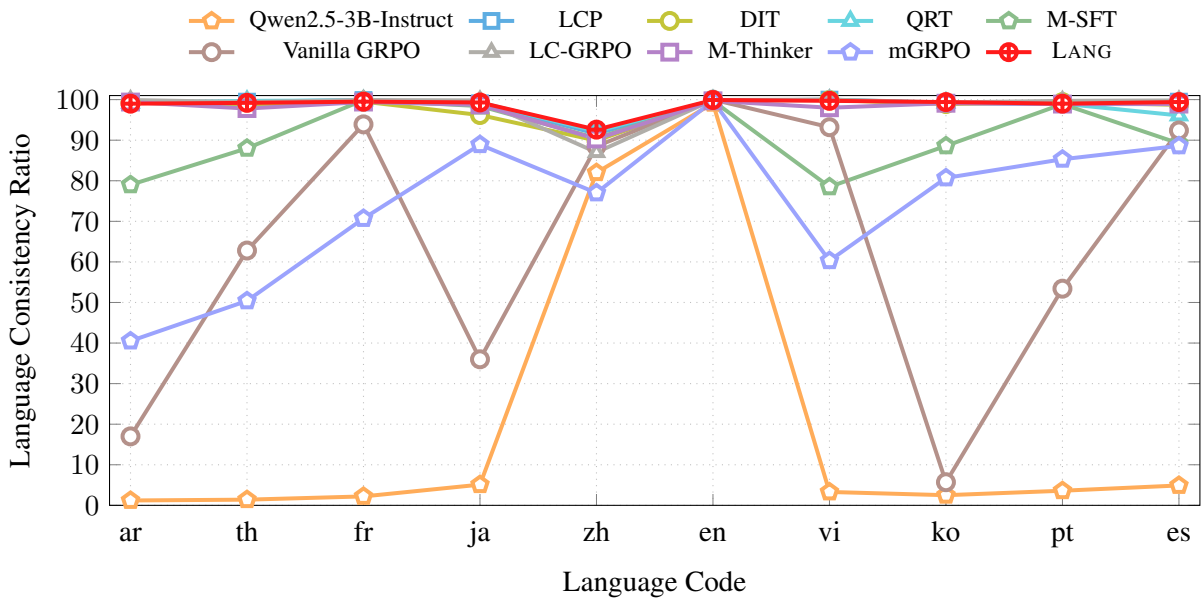

Figure 11: The input-output language consistency ratio of different methods for each language on MMATH test sets with Qwen2.5-3B-Instruct model.

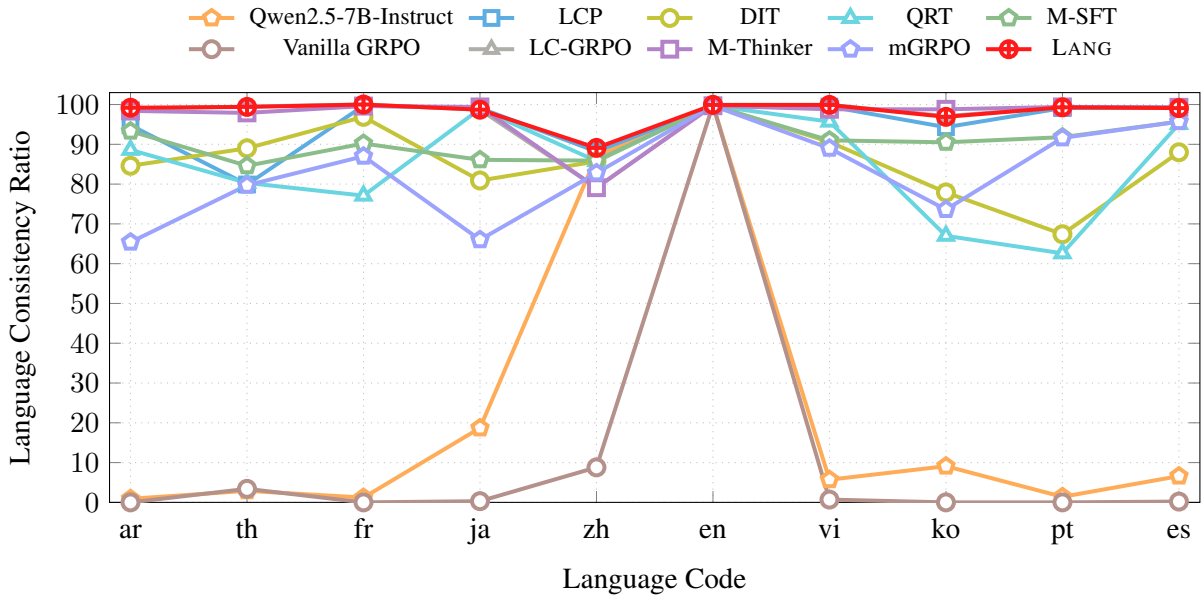

Figure 12: The input-output language consistency ratio of different methods for each language on MMATH test sets with Qwen2.5-7B-Instruct model.

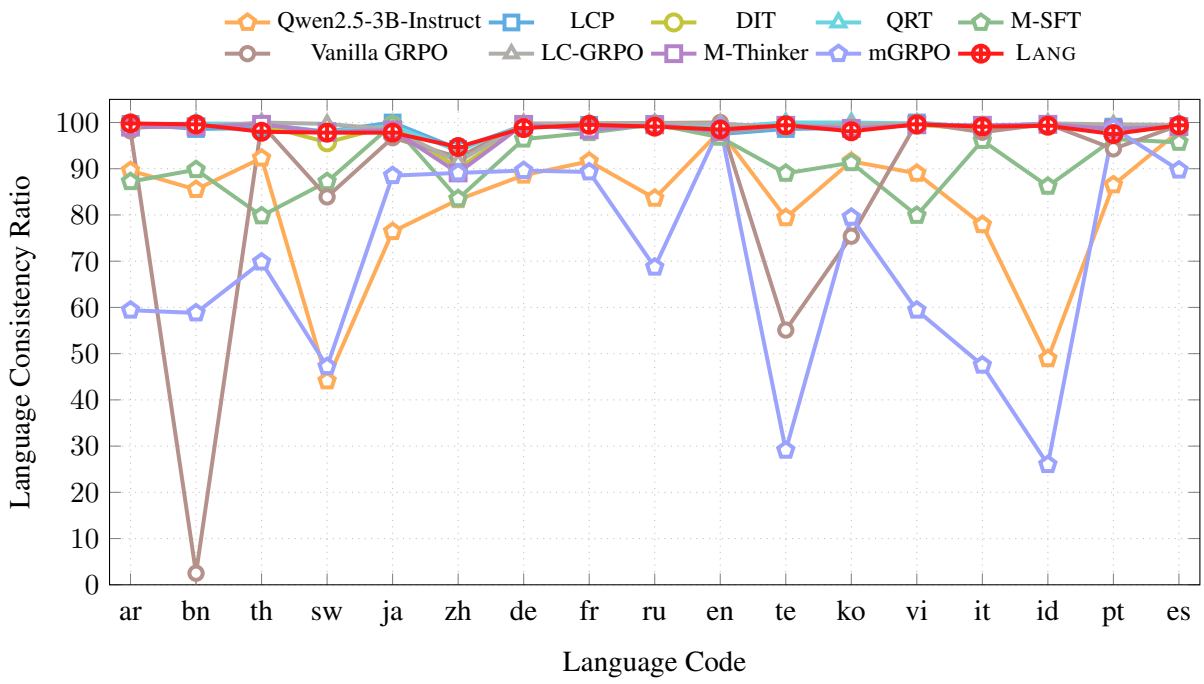

Figure 13: The input-output language consistency ratio of different methods across languages and levels on PolyMath with Qwen2.5-3B-Instruct model.

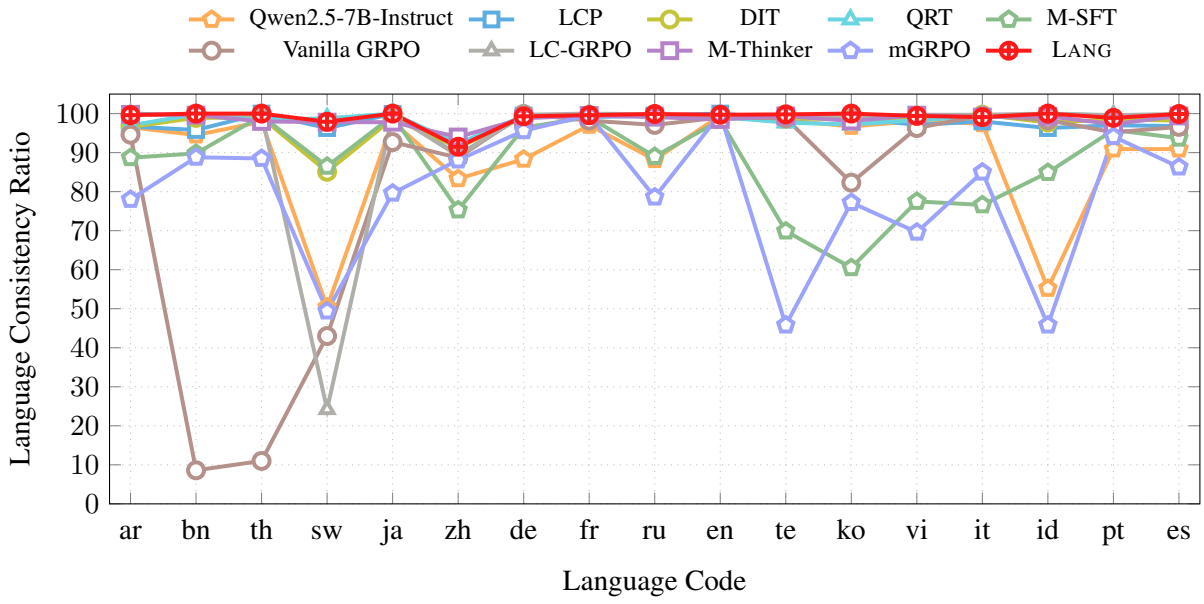

Figure 14: The input-output language consistency ratio of different methods across languages and levels on PolyMath with Qwen2.5-7B-Instruct model.