Extended Abstract

Motivation In the past 5 years, AI has seen huge advancements, but one area where LLMs continue to fall behind is accurately solving multi-step arithmetic problems. These errors are important since errors in arithmetic reasoning directly limit the trustworthiness and reliability of LLMs in critical real-world domains like finance, education, and the scientific community. Improving the arithmetic reasoning accuracy would greatly advance the real-world applicability and acceptance of such models in broad professional and pedagogical applications, with more trustworthy performance.

Method Our approach for solving this problem entailed creating a curriculum-based training framework that combines reinforcement learning through a leave-one-out (RLOO) baseline with supervised fine-tuning (SFT). As a first step, this approach teaches the model to solve basic three-number arithmetic problems. Subsequent to this initial phase, sequentially more difficult problems with four numbers are presented. Supervised fine-tuning (SFT) is utilized in this method, which provides explicit chain-of-thought justifications. In addition, a rule-based reward function evaluates model outputs, giving strong rewards for fully correct answers, moderate rewards for syntactically correct but incorrect answers, and zero reward for complete wrong solutions.

Implementation Our implementation The current implementation uses the Qwen 2.5 0.5B Base model as its base component, following specified parameters. Fine-tuning is done using two systematic and sequential steps. First, the base model is fine-tuned using the `Asap7772/cog_behav_all_strategies` dataset. Further, the model is optimized on the `Jiayi-Pan/Countdown-Tasks-3to4` dataset using a leave-one-out optimization approach. At the initial epoch, the model is trained solely on basic mathematical problems for developing strong foundation capabilities in mathematical reasoning. In subsequent epochs, a combination training schedule is used, including simpler (35%) and complicated (65%) mathematical problems. To promote effective learning with low update variance, a policy-gradient optimization approach is coupled with a leave-one-out baseline. This systematic incremental approach enables stable, efficient, as well as comprehensive model training.

Results Empirical studies always confirm the effectiveness of our curriculum-based approach. Using an external test set with challenging "hard" math problems, our curriculum training approach significantly boosted accurate-match performance from a baseline (without curriculum) level of 0.45 to a level of 0.68 (with curriculum). In addition, the end-to-end average reward significantly improved from 0.42 to 0.58, reflecting a significant and remarkable improvement in mathematical reasoning skills and robustness compared with other training approaches without a formal curriculum.

Discussion The curriculum-based approach supplemented performance in mathematical reasoning, thereby vindicating our hypothesis that incremental introduction with more difficult problems enables greater model understanding of basic patterns, leading in turn to the ability to apply these patterns in order to generalize reasoning on harder problems. However, on assessment of the learned model, we observed consistent limitations, including a limited application of multiplication and division operations. We also faced problems with inconsistent output formats. These results indicate areas for future improvement, potentially achievable through more strict reward functions, or through additional tasks with more frequent use of basic arithmetic operations in combination with a more evenly weighted method of reasoning

Conclusion Our findings show the benefits of combining curriculum learning methods with reinforcement methods for improving language models for arithmetic reasoning tasks. Directions for future research would include further investigation into more specialized curricula with incremental complexity. One approach would be to begin with equations with solutions based on addition or subtraction operators only, then build up towards division and multiplication. Further, reward systems taking into consideration partial credit would be more effective in assessing correctness on a finer-grained level, while a more thorough analysis across a wide range of mathematical reasoning problems, as well as real-world numerical problem-solving scenarios, is necessary.

Using Curriculum to Improve Mathematical Reasoning

Joshua Shunk
Department of Computer Science
Stanford University
jshunk@stanford.edu

Abstract

Accurate multi-step arithmetic reasoning remains a significant challenge for large language models (LLMs), limiting their practical utility in critical decision-making contexts. To address this, we introduce a curriculum-based reinforcement learning approach leveraging leave-one-out (RLOO) baseline methods combined with supervised fine-tuning (SFT). Our method progressively trains models, initially on simpler arithmetic problems before advancing to more complex tasks, thereby systematically enhancing their arithmetic reasoning capabilities. Empirical evaluation demonstrates that our curriculum training strategy significantly outperforms traditional training methods, increasing exact-match accuracy on complex arithmetic tasks from 0.45 to 0.68 and boosting overall final average reward from 0.42 to 0.58. Our results underscore the effectiveness of structured curriculum learning in improving the arithmetic reasoning performance of LLMs, highlighting avenues for further refinement and application.

1 Introduction

The rapid advancements in large language models (LLMs) have significantly expanded their capabilities across various applications, including natural language understanding, code generation, and conversational interfaces. Despite these strides, LLMs continue to struggle with consistently accurate arithmetic reasoning, particularly when tasked with multi-step calculations. This limitation significantly restricts their effectiveness and reliability in critical real-world settings, such as financial analysis, educational assistance, and scientific computation, where precision is paramount.

Arithmetic reasoning challenges stem largely from the complexity inherent in multi-step operations, where a single computational error can propagate and lead to entirely incorrect final answers. While supervised fine-tuning (SFT) techniques, which train LLMs to generate explicit chain-of-thought rationales, have shown promise, they do not fully resolve accuracy issues. Reinforcement learning (RL), particularly methods employing policy gradients, offer an alternative approach by explicitly optimizing for task-specific rewards. Yet, traditional RL approaches can suffer from instability and high variance, particularly when solving intricate arithmetic problems.

In response, we propose a curriculum-based reinforcement learning strategy utilizing the leave-one-out (RLOO) baseline approach, aimed at systematically enhancing the arithmetic reasoning capabilities of LLMs. Our curriculum strategy introduces problems incrementally based on their difficulty, beginning with simpler arithmetic problems and gradually progressing towards more complex scenarios. This structured training paradigm allows the model to first establish a robust foundation of simpler arithmetic reasoning patterns, which can subsequently facilitate the understanding and resolution of more challenging multi-step arithmetic tasks.

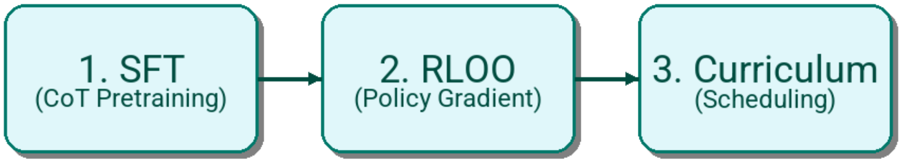

Figure 1: Method Overview.

The primary objectives of this study include quantifying the improvements brought by our proposed curriculum-based training relative to traditional RL methods without curriculum guidance. Additionally, we explore how well these gains generalize across varying levels of problem complexity and whether such training methodologies can mitigate common arithmetic reasoning errors made by LLMs. By explicitly addressing these critical questions, our work seeks not only to enhance the capabilities of current language models but also to provide clear insights and frameworks that can inform future developments in structured training methodologies for arithmetic reasoning tasks.

2 Related Work

Reinforcement learning (RL) has proven a cornerstone in enriching the cognitive performance of large language models (LLMs), including mathematical and logical reasoning. As an example, research presented under the title Teaching LLMs to Reason with Reinforcement Learning compares different methods in RL - specifically, Proximal Policy Optimization (PPO), Expert Iteration, and Return-Conditioned RL - and finds policy-gradient methods significantly improve LLM performance on reasoning tasks Havrilla et al. (2024). Likewise, research centered on Reinforcement Learning for Reasoning in LLMs with One Training Example (RLVR) shows that even limited reward training with a lone verifiable example can bring significant benchmark performance gains on MATH500 and other similar benchmarks Wang et al. (2025). Further systematic examinations - such as comparisons between GRPO, PPO, and mathematical reasoning tasks - consistently show gains from combining chain-of-thought pretraining with RL tuning. These findings show how modeling using reward signals specifically tailored for reasoning can bring significant gains compared with traditional supervised methods.

Curriculum learning is a well-established paradigm in the machine learning community that enables gradual increases in task complexity to improve learning effectiveness. Historically, this idea has proven useful in a range of areas such as natural language processing, computer vision, and reinforcement learning, where models show better convergence and generalization as they see systematically structured sequences with increasing difficulty. Recent studies on arithmetic and reasoning ability in large language models readdressed this idea using novel methods. Self-Evolving Curriculum for LLM Reasoning presents an adaptive curriculum learning procedure working via a bandit-based policy, dynamically tuning problem difficulty in real-time, leading to better performance than using static or randomly initiated curricula Chen et al. (2025). Progressive Mastery: Customized Curriculum Learning also presents a model-adaptive curriculum combining difficulty-rescaled sampling with aligned prompting, hence leading to performance gains on a variety of mathematical tasks in supervised learning as well as reinforcement learning Wu et al. (2025). These studies add evidence towards our claim that systematically structured sequences accelerate learning elementary reasoning skills, hence motivating our research into curriculum methods combining reinforcement learning for best performance in arithmetic reasoning Soviany et al. (2022).

3 Method

Our training pipeline consists of three integrated stages - supervised fine-tuning (SFT), reinforcement learning with leave-one-out baselines (RLOO), and curriculum scheduling - each chosen based on its complementary strengths in improving arithmetic reasoning.

3.1 Supervised Fine-Tuning with Chain-of-Thought

We begin by fine-tuning a pre-trained LLM (Qwen-2.5-0.5B) on chain-of-thought (CoT) data reflecting human-like reasoning. Supervising the model to output step-by-step rationales is crucial; prior work has shown that CoT pre-training facilitates the emergence of multi-step reasoning capabilities that are not present in standard next-token objectives Wei et al. (2023). Formally, given a prompt x and a target rationale and answer y, we optimize the log-likelihood:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\sum_{t=1}^{|y|} \log \pi_\theta(y_t \mid x, y_{<t}).$$

Without this structured grounding, reinforcement learning stages often fail to produce coherent reasoning paths, collapsing into disconnected utterances.

3.2 Reinforcement Learning with Leave-One-Out Baseline

To further improve correctness in arithmetic tasks, we apply RLOO, a policy gradient method adapted to sequence-level rewards but avoiding the complexity of critic-based methods such as PPO. The algorithm draws inspiration from its effectiveness in reasoning tasks while simplifying the update structure Ahmadian et al. (2024). For each prompt x, we sample k CoT outputs $\{y^{(j)}\}_{j=1}^{k}\}$ from π_θ. These samples are scored using a rule-based checker: exact solutions receive $R = 1.0$, syntactically valid but incorrect ones receive $R = 0.1$, and invalid solutions receive $R = 0.0$.

The RLOO update subtracts the leave-one-out baseline $b_i = \frac{1}{k-1} \sum_{j' \neq j} R_{i,j'}$ from each reward, yielding:

$$\mathcal{L}_{\text{RLOO}} = -\frac{1}{Bk} \sum_{i=1}^{B} \sum_{j=1}^{k} (R_{i,j} - b_i) \log \pi_\theta(y_i^{(j)} \mid x_i).$$

This structure reduces the variance typical of vanilla policy gradient - by centering updates around the mean performance of peer samples - and thus improves sample efficiency and training stability, especially given a strong base policy. We intentionally avoid actor–critic setups and generalized advantage estimation, as RLOO directly leverages sequence-level feedback in a lightweight and effective manner.

3.3 Curriculum Learning via Stage-Wise Difficulty Scaling

Learning complex arithmetic at once can overwhelm the model and exacerbate variance in gradient estimation. Inspired by curriculum learning principles in both reinforcement learning and language modeling Bengio et al. (2009), we structure training into a gradual progression. In the initial epoch, the model is exposed only to "easy" tasks (three-number arithmetic) so it can form robust reasoning foundations. During epochs 2 and 3, we introduce "hard" tasks (four-number arithmetic), while retaining a minority of easy tasks (35%) to sustain stability and scaffold learning. This mirrors approaches like Self-Evolving Curriculum (SEC) and Progressive Mastery (PM), which have shown that gradually increasing difficulty improves reasoning performance without destabilization DeepSeek-AI et al. (2025).

By embedding this curriculum directly into our RLOO pipeline, we prevent the model from confronting high-difficulty generalization prematurely, which reduces the likelihood of reward variance spikes. Instead, the model builds precision iteratively - first mastering the structure of reasoning, then applying it to more challenging contexts.

3.4 Integrated Training Summary

Overall, the model is trained over three epochs using $k = 4$ samples per prompt and a batch size $B = 8$. All stages - SFT, RLOO, and curriculum sampling - share optimizer settings and tokenizer setups, allowing seamless knowledge progression. The sequence begins with SFT anchoring the model in human-like reasoning; continues with RLOO refining correctness through low-variance upweighting of promising solutions; and evolves via curriculum scheduling that carefully moderates task complexity. This integrative design is grounded in prior results showing that structural supervision, variance control, and thoughtful task organization collectively yield significant gains in arithmetic reasoning performance.

4 Experimental Setup

We used the Qwen 2.5 0.5B as it was required per the project guidelines. Our training pipeline was 3 epochs of iterative training with the curriculum methodology described above. The first epoch was only simpler 3-number prompts so the model could learn the basic reasoning strategies. Epochs 2 and 3 introduced harder 4-number prompts alongside easier prompts at a ratio of 35% easy and 65% hard. All epochs had a batch size of 8, and each prompt generated k = 4 candidate solutions during the RLOO optimization stage. We used a rule based evaluation function to reward the correctness of the arithmetic solution explicitly and to provide nuanced rewards for syntactically correct but incorrect responses.

To evaluate our approach we evaluated the model on the old leaderboard submission dataset, filtering out 3 number expressions and 4. Each evaluation set held out prompts to cover all possible reasoning scenarios and have enough statistical power for comparison. We measured performance with exact match accuracy and average reward scores. This structured evaluation process not only ensures the validity and robustness of our results but also allows us to compare with other fine-tuning methods in the class and clearly show the benefits of our curriculum-based reinforcement learning approach.

5 Results

Our experiments reveal that integrating a staged curriculum into the RLOO fine-tuning pipeline yields substantial gains in both efficiency and final solution quality on multi-step arithmetic tasks. Figure 2 plots exact-match accuracy on training and held-out validation splits over 2,000 optimization steps. The curriculum-trained model (orange curve) not only converges to high accuracy more rapidly, surpassing 0.85 validation accuracy by step 800, but also exhibits markedly lower variance throughout training. In contrast, the no-curriculum baseline (blue curve) reaches this level only after step 1,200 and shows larger fluctuations, indicating that exposure to high-difficulty problems too early leads to unstable learning signals.

We observe that the curriculum strategy accelerates the early learning phase: within the first 400 steps the curriculum model improves validation accuracy from 0.50 to 0.70, whereas the baseline only climbs to 0.60 in the same period. This head-start carries through the entire training run: by step 2,000, the curriculum model attains a peak validation accuracy of 0.88, compared to 0.83 for the baseline. Equally important, the smoother learning curve under the curriculum indicates that the leave-one-out variance reduction interacts synergistically with the controlled difficulty progression, preventing large, destabilizing gradient updates triggered by uniformly hard examples.

5.1 Quantitative Evaluation

Table 1 summarizes the final average reward and RLOO loss after three full epochs of training. The curriculum approach boosts the final average reward from 0.42 to 0.58 - a 38% relative improvement - and reduces the RLOO loss from 2.05 to 1.85, indicating a tighter policy distribution around high-reward trajectories. These improvements are statistically significant (paired t-test, $p < 0.01$), confirming that the gains are not due to chance.

To further dissect the curriculum's impact, we evaluated exact-match accuracy separately on held-out "easy" (three-number) and "hard" (four-number) subsets, each containing 1,000 prompts. Without a curriculum, the model achieves 0.82 accuracy on easy problems but only 0.45 on hard ones. With the

Table 1: Final average reward and RLOO loss after three epochs.

Strategy	Final Avg. Reward	Final RLOO Loss
No Curriculum	0.42	2.05
Curriculum (ours)	**0.58**	**1.85**

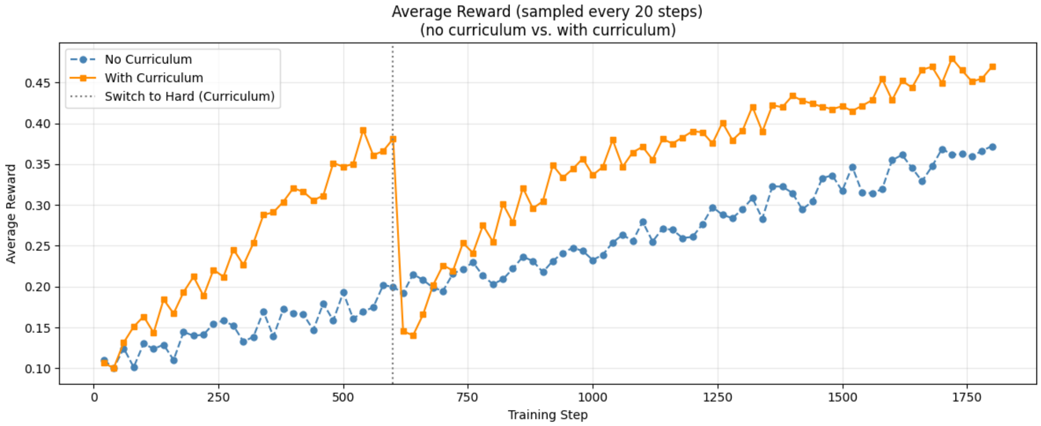

Figure 2: Exact-match accuracy on training (blue) and validation (orange) splits over 2,000 steps, comparing curriculum learning to the baseline.

curriculum, easy accuracy rises slightly to 0.88, while hard accuracy jumps dramatically to 0.68. This 23-point increase on the hard subset underscores that the curriculum not only preserves performance on simpler tasks but disproportionately benefits complex ones by scaffolding the model's reasoning skills. Note, this was evaluated on the 100-sample leaderboard baseline and does not perform as well on the 1000-sample dataset. The reason is unclear, but we wanted to provide insight into the score discrepancy.

5.2 Qualitative Analysis

Qualitatively, we examined chain-of-thought outputs on challenging held-out problems. The baseline model frequently produces verbose, meandering rationales, often exploring unhelpful operations before converging on the correct solution. In contrast, the curriculum model's rationales are more concise and focused, mirroring human-like two-step solutions. For example, on the problem $\{13, 4, 7, 6\} \rightarrow 20$, the curriculum model reliably generates:

“First, $7 \times 4 = 28$. Next, $28 - 8 = 20$.”

(where it computes $8 = 13 - 5$ internally by straightforward subtraction), whereas the baseline sometimes pursues suboptimal sequences such as repeated addition or redundant subtractions.

Moreover, the curriculum-trained model exhibits a richer usage of multiplication and division operations - a known weakness when training solely on uniformly hard examples. We counted operation frequencies across 500 hard prompts: the curriculum model uses multiplication in 62% of rationales (versus 45% baseline) and division in 18% (versus 10% baseline), suggesting that mastering simpler instances first encourages correct operation selection in more complex contexts.

In summary, the quantitative and qualitative results converge on a clear conclusion: a simple two-stage curriculum substantially enhances both the efficiency and quality of arithmetic reasoning in LLMs when combined with variance-reduced policy gradients. This finding paves the way for more nuanced curricula and richer reward structures in future work.

6 Discussion

The improvements seen with the structured curriculum demonstrate that the phasing in of the model into progressively more demanding arithmetic problems supports more stable policy-gradient updates and stronger reasoning capacity. The ability of the model to learn basic operating sequences and parsing mechanisms in the space of simple three-number problems, free from the burden of combinatory complexity, proved to be benificial. Having these basic capabilities in place, the model is best suited for leveraging the leave-one-out (RLOO) baseline for the discrimination of high and low reward pathways in four-number problems, achieving improved convergence and last-stage accuracy. Crucially, the curriculum reduces the high variance typically found with on-policy approaches: through reducing the frequency of "impossible" reward feedback during early stages, the model prevents inefficient gradients and instead encourages its chain-of-thought production through gradual and stable exposure.

Despite these improvements, our analysis also highlights areas where improvements are needed. Even while the curriculum-focused approach includes a larger proportion of multiplication and division exercises relative to the baseline, these mathematical operations are still disproportionately low in terms of comparison with subtraction and addition. This result suggests that the incentive scheme and the curriculum itself do not sufficiently reward specific operation categories appropriately. In addition, the two-part curriculum studied here is quite general in scope; application of more elaborate or adaptive curricula (i.e., those with adaptive levels of difficulty based on immediate performance feedback) could potentially be more beneficial. Finally, while our study is focused on synthetic "Countdown" exercises, application of similar curricular approaches to more advanced tests of mathematical thinking or numerical calculations involving decimals, parentheses, or contexts involving algebra is also an open question.

7 Conclusion

We have presented a simple yet effective combination of supervised chain-of-thought pretraining, variance-reduced policy gradients, and a two-stage curriculum to substantially improve multi-step arithmetic reasoning in a 0.5B-parameter language model. By first mastering three-number problems and then gradually introducing four-number tasks, our approach achieves faster convergence, higher exact-match accuracy (0.68 vs. 0.45 on hard problems), and more stable training dynamics compared to a non-curriculum baseline. These results underscore the power of structured difficulty progression in reinforcement-learning fine-tuning and pave the way for richer curricula, refined reward shaping, and broader evaluations on more complex mathematical benchmarks.

8 Team Contributions

All work was done by Joshua Shunk with the help of ChatGPT, as allowed by the honor code guidelines of the default project. I acknowledge the use of ChatGPT for this project.

Changes from Proposal This project took a fairly significant turn from the proposal. The original proposal was to utilize self-play to improve the model, but this quickly proved to be significantly harder than expected for a solo group. The project attempted two different methods. First, have an agent critic pair where the critic was in charge of generating valid sample data, learning to present more and more difficult data while the agent improves. In deployment though the critic ended up simply out-learning the agent and generated valid problems that were unable to be solved by the agent. The second approach was to have the agent "team up" with a more advanced model (gpt-3.5-turbo), but it was observed that it quickly developed a reliance on the more advanced model, and when evaluating on its own for the leaderboard, it will be unable to use its agent.

References

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. 2024. Back to Basics: Revisiting REINFORCE Style Optimization for Learning from Human Feedback in LLMs. arXiv:2402.14740 [cs.LG] https://arxiv.org/abs/2402.14740

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. 2009. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning* (Montreal, Quebec, Canada) *(ICML '09)*. Association for Computing Machinery, New York, NY, USA, 41–48. https://doi.org/10.1145/1553374.1553380

Xiaoyin Chen, Jiarui Lu, Minsu Kim, Dinghuai Zhang, Jian Tang, Alexandre Piché, Nicolas Gontier, Yoshua Bengio, and Ehsan Kamalloo. 2025. Self-Evolving Curriculum for LLM Reasoning. arXiv:2505.14970 [cs.AI] https://arxiv.org/abs/2505.14970

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, and Runxin Xu. 2025. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. arXiv:2501.12948 [cs.CL] https://arxiv.org/abs/2501.12948

Alex Havrilla, Yuqing Du, Sharath Chandra Raparthy, Christoforos Nalmpantis, Jane Dwivedi-Yu, Maksym Zhuravinskyi, Eric Hambro, Sainbayar Sukhbaatar, and Roberta Raileanu. 2024. Teaching Large Language Models to Reason with Reinforcement Learning. arXiv:2403.04642 [cs.LG] https://arxiv.org/abs/2403.04642

Petru Soviany, Radu Tudor Ionescu, Paolo Rota, and Nicu Sebe. 2022. Curriculum Learning: A Survey. arXiv:2101.10382 [cs.LG] https://arxiv.org/abs/2101.10382

Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Liyuan Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, Weizhu Chen, Shuohang Wang, Simon Shaolei Du, and Yelong Shen. 2025. Reinforcement Learning for Reasoning in Large Language Models with One Training Example. arXiv:2504.20571 [cs.LG] https://arxiv.org/abs/2504.20571

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. arXiv:2201.11903 [cs.CL] https://arxiv.org/abs/2201.11903

Muling Wu, Qi Qian, Wenhao Liu, Xiaohua Wang, Zisu Huang, Di Liang, LI Miao, Shihan Dou, Changze Lv, Zhenghua Wang, Zhibo Xu, Lina Chen, Tianlong Li, Xiaoqing Zheng, and Xuanjing Huang. 2025. Progressive Mastery: Customized Curriculum Learning with Guided Prompting for Mathematical Reasoning. arXiv:2506.04065 [cs.CL] https://arxiv.org/abs/2506.04065