

LIMITS OF DIFFICULTY SCALING: HARD SAMPLES YIELD DIMINISHING RETURNS IN GRPO-TUNED SLMs

Suraj Yadav, Siddharth Yadav*, Parth Goyal*

IIIT Delhi

{suraj24098, siddharth23525, parth23371}@iiitd.ac.in

ABSTRACT

Recent alignment work on Large Language Models (LLMs) suggests preference optimization can improve reasoning by shifting probability mass toward better solutions. We test this claim in a resource-constrained setting by applying GRPO with LoRA to SLMs (0.5B–3B) for math reasoning on GSM8K and MATH datasets with difficulty-stratified analyses. As problem difficulty increases, accuracy plateaus, revealing a capacity boundary: GRPO primarily reshapes output preferences without reliably improving hardest-tier solving. Consistent with this, training GRPO only on lower-difficulty problems matches full-dataset accuracy across difficulty tiers while using only $\sim 45\%$ training steps, indicating diminishing returns from harder samples in this regime. We also find a cross-dataset generalization effect: GSM8K-trained GRPO achieves higher accuracy on the numeric subset of MATH than MATH-trained GRPO, exceeding it by $\sim 5\%$ at 1.5B and by $\sim 3\%$ at 3B. We show that the best achievable gains depend strongly on the base model’s prior reasoning competence and the dataset’s difficulty profile.

1 INTRODUCTION

Group Relative Policy Optimization (GRPO) Shao et al. (2024), has proven effective for enhancing mathematical reasoning in large language models (LLMs). However, while SLMs often inherit reasoning via distillation from larger teachers Guo et al. (2025), the efficacy of applying GRPO directly in resource-constrained regimes remains under-explored. Current literature generally suggests that exposure to higher-difficulty training data is associated with improved reasoning performance Tong et al. (2024), particularly in LLMs. In compute resource constraint settings, it is critical to determine if Small Language Models (SLMs) possess the intrinsic capacity Wei et al. (2022) to benefit from complex alignment samples, or if they face a capacity boundary.

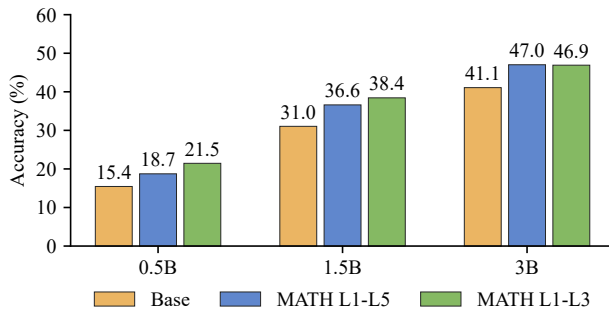


Figure 1: Comparison of accuracy on the MATH dataset for Qwen2.5 0.5B, 1.5B, and 3B models Qwen et al. (2025). Models trained on the lower-difficulty subsets (L1–L3) achieve comparable or superior performance to those trained on the full dataset (L1–L5).

Capacity boundary is a threshold below which algorithmic improvements cannot overcome the fundamental lack of model expressivity required for complex reasoning.

*Equal contribution.

In this work, we test this hypothesis by conducting a difficulty-stratified Ding et al. (2025) Gao et al. (2025) analysis of GRPO on SLMs. We fine-tune models ranging from 0.5B to 3B parameters using LoRA Hu et al. (2021) on the GSM8K Wei et al. (2022) and MATH Tong et al. (2024) datasets. Unlike standard evaluations, we perform a difficulty-stratified analysis, isolating the impact of training data complexity on downstream performance across different tiers of problem hardness.

1. **Identification of the Capacity Boundary:** We provide empirical evidence that GRPO-based alignment hits a hard saturation point in SLMs, preventing further learning on complex reasoning tasks.
2. **Efficiency via Difficulty Stratification:** We demonstrate that filtering out "hard" training samples does not degrade performance. Training on only the lower-difficulty $\sim 45\%$ of the dataset achieves competitive results with full-dataset training, offering a protocol for significantly more efficient alignment.
3. **Cross-Dataset Generalization Analysis:** We analyze the transfer capabilities between models trained on GSM8K and MATH, identifying cases where "easier" training distributions (GSM8K) provide a stronger signal for numeric reasoning than domain-specific "hard" distributions (MATH).

2 RELATED WORK

2.1 POST-TRAINING ALIGNMENT AND GRPO

Post-training alignment methods like RLHF Ouyang et al. (2022), DPO Rafailov et al. (2024), and specifically Group Relative Policy Optimization (GRPO) Shao et al. (2024) effectively enhance reasoning in Large Language Models but remain under-explored for Small Language Models (SLMs). Unlike traditional RLHF, which relies on a separate, memory-intensive value function to compute policy baselines, GRPO samples a group of G outputs $\{o_1, \dots, o_G\}$ for a single prompt q . It then computes the baseline directly from this group’s empirical mean. The optimization objective is formulated as:

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{q \sim \mathcal{D}, o \sim \pi_{\theta_{\text{old}}}} \left[\frac{1}{G} \sum_{i=1}^G \frac{1}{|o_i|} \sum_{t=1}^{|o_i|} \min(r_{i,t}(\theta) \hat{A}_{i,t}, \text{clip}(r_{i,t}(\theta), 1 - \epsilon, 1 + \epsilon) \hat{A}_{i,t}) - \beta D_{\text{KL}}(\pi_{\theta} \parallel \pi_{\text{ref}}) \right] \quad (1)$$

where

$$r_{i,t}(\theta) = \frac{\pi_{\theta}(o_{i,t} \mid q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t})}.$$

where the advantage $\hat{A}_{i,t}$ is the normalized reward of output o_i relative to its group. This mechanism significantly reduces memory overhead and naturally incentivizes the model to outperform its own average generation, a dynamic well-suited for iteratively improving mathematical Chain-of-Thought reasoning.

2.2 DATA COMPLEXITY AND THE CAPACITY BOUNDARY

Furthermore, while existing curricula prioritize high-complexity data Ding et al. (2025), we demonstrate that for capacity-constrained SLMs, such strategies often induce negative transfer. Our work investigates this "capacity boundary," establishing that simpler signals yield superior generalization in sub-3B regimes. (See Appendix A for extended review).

3 METHODOLOGY

Training Framework We adopt a two-stage training protocol. Models Qwen 2.5 0.5B, 1.5B, 3B Qwen et al. (2025) first undergo a brief phase of Supervised Fine-Tuning (SFT) to establish instruction-following behavior and baseline reasoning capabilities. Subsequently, we align these

SFT-initialized models using Group Relative Policy Optimization (GRPO) via the TRL library von Werra et al. (2020). To ensure parameter efficiency, we apply Low-Rank Adaptation (LoRA) to all linear projection layers utilizing the PEFT library Mangrulkar et al. (2022). Detailed hyperparameters for both SFT and GRPO stages are provided in Appendix B.

Difficulty-Aware Reward Modeling We design a composite reward function that enforces both strict XML-based formatting and reasoning correctness. A key component of our methodology for the MATH dataset is difficulty scaling: we dynamically weight correctness rewards based on the problem’s difficulty level. This ensures that successful reasoning on complex tasks provides a stronger optimization signal.

Dataset & Stratification We utilize the GSM8K dataset and a targeted selection of the Hendrycks MATH benchmark, specifically focusing on four core domains: *algebra*, *counting.and.probability*, *geometry*, and *number.theory*. To verify our hypothesis regarding capacity boundaries, we map these into a unified two-tier complexity hierarchy:

- **Low Difficulty:** GSM8K (grade-school arithmetic), MATH Levels 1–3.
- **High Difficulty:** MATH Levels 4–5.

Evaluation Protocol We evaluate all models using greedy decoding ($T = 0$) to assess deterministic reasoning capabilities. We benchmark performance improvements directly against the **SFT-tuned baselines** (the model state prior to GRPO alignment). Our critical analysis focuses on **upward transfer**: measuring whether models trained exclusively on "Low" data partitions can generalize to "High" difficulty test samples without direct exposure to them. Additionally, we extend this evaluation to both GSM8K-trained and MATH-trained models, testing them on the numerical-answer subset of the full MATH test set (see Appendix B.2).

4 EXPERIMENTAL RESULTS

4.1 THE CAPACITY BOUNDARY: SATURATION ON HIGH-DIFFICULTY TASKS

We investigate whether exposing small models to high-difficulty reasoning paths improves general performance. Figure 1 compares models tuned on the full MATH dataset (Levels 1–5) versus the easier subset (Levels 1–3). We observe that including high-difficulty problems yields **no significant gain** across any model size. In fact, for the 0.5B and 1.5B models, the inclusion of Level 4–5 data proves detrimental: the 0.5B model achieves **21.5% accuracy** on the L1–L3 subset but drops to **18.7%** when trained on the full dataset. Similarly, the 1.5B model sees a regression from 38.4% to 36.6% with the addition of harder tasks. Even for the 3B model, performance saturates, showing negligible difference between conditions (46.9% vs. 47.0%). Crucially, to verify that this saturation is not merely an optimization artifact of parameter-efficient fine-tuning, we conducted a full fine-tuning (FFT) ablation on the 0.5B model. The FFT results mirrored the LoRA findings, showing a similar trend of performance when high-difficulty samples were included (detailed in Appendix F). This confirms a strict capacity boundary where small models cannot effectively leverage rewards from problems beyond their reasoning horizon. As detailed in Appendix C Figure 4 and Figure 5, we observe distinct failure mechanisms dictated by model scale. The 0.5B model simply lacks the capacity to sustain complex inference; it fails to reason deeply, with its average generation length plateauing at approximately **330 tokens** for Level 5 problems. In contrast, the 1.5B and 3B models actively attempt to solve these tasks, scaling their chain-of-thought to nearly **400 tokens** for Level 5 problems. However, this extended reasoning leads to *token overflow*: the models frequently "get lost" in their own convoluted reasoning traces or hit length constraints before a final answer can be extracted (See Appendix H for extended review).

4.2 DIMINISHING RETURNS: HARD DATA AS COMPUTATIONAL INEFFICIENCY

Given the capacity boundary, we investigate the utility of including hard samples in the training distribution. Figure 2 compares models trained on the Full Dataset (MATH L1–L5) versus those trained exclusively on the simple subset (MATH L1–L3). Remarkably, we observe a signal-to-noise degradation where training on high-difficulty problems degrades performance on those very same

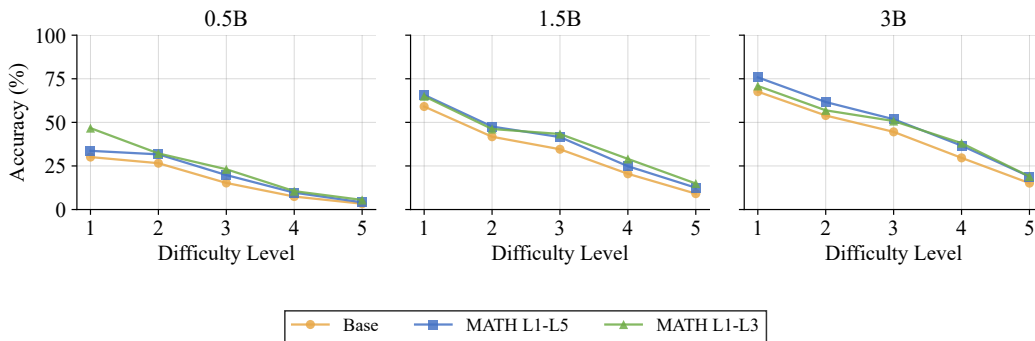


Figure 2: Difficulty-stratified MATH accuracy for base and GRPO-tuned models. GRPO trained on full data (L1–L5) performs similarly to GRPO trained on L1–L3, suggesting limited benefit from including higher-difficulty problems (especially for smaller models).

problems for smaller models. For the 1.5B model, restricting training to the easy subset (L1–L3) outperforms training on the full dataset (L1–L5) on **Level 4** tasks (**29.0%** vs. **24.9%**) and **Level 5** tasks (**14.9%** vs. **12.5%**). To ensure this performance gap is driven by the exclusion of hard samples rather than simply a reduction in dataset size, we trained a control 1.5B model on a random subsample of the full dataset, exactly matched to the size of the L1–L3 split. This size-matched random subset performed even worse than the full dataset, confirming that the presence of high-difficulty samples not the reduced training duration is the root cause of the degradation (detailed in Appendix G). This suggests that for models below the 3B threshold, unsolvable L5 samples do not provide a learning signal; instead, they introduce high-variance gradients as evidenced by the volatile reward trajectories in Appendix E Figure 7 that actively interfere with the consolidation of intermediate reasoning patterns. For the 3B model, the trade-off is distinct: while the Full Dataset yields higher accuracy on Level 1 (75.8% vs. 70.9%), the Easy-Only curriculum still achieves superior generalization to Level 4 difficulty (**38.1%** vs. **36.7%**). Consequently, by filtering these hard samples, we achieve superior alignment outcomes on complex tasks while using only approximately 45% of total training steps.

4.3 CROSS-DATASET GENERALIZATION: SIMPLER SIGNALS TRANSFER BETTER

We observe a cross-dataset generalization effect in which training on the simpler GSM8K dataset can transfer better than training on MATH itself. Figure 3 reports the evaluation on the *Numerical* subset of MATH.

The transfer from GSM8K improves with scale: at 1.5B, GSM8K-trained model reaches **44.6%** accuracy vs. **39.4%/39.0%** for the (MATH L1–L3/L1–L5) models, and at 3B it achieves **54.1%** vs. **50.4%/51.2%**. Conversely, the 0.5B model does not benefit from this transfer (**16.3%** on GSM8K vs. **22.1%/20.2%** on MATH), suggesting that very small models need target-dataset supervision (even when both datasets are math-specific). Appendix D Table 5 indicates that these gains come from cleaner transfer in foundational topics. For instance, in **Algebra**, the GSM8K-trained 1.5B model outperforms the MATH-trained baseline across difficulty (e.g., **91.2%** vs. **81.6%** at Level 1; **27.3%** vs. **21.7%** at Level 5), which is consistent with GSM8K providing a lower-variance numeric signal than MATH’s heterogeneous objectives.

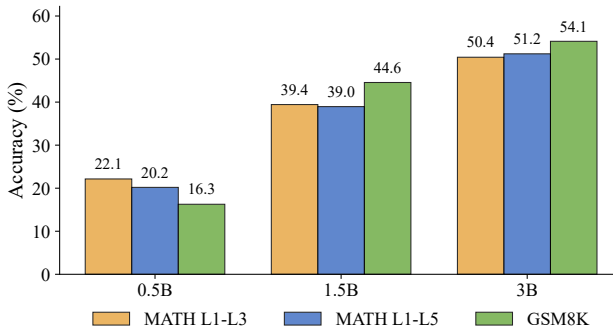


Figure 3: **Cross-dataset generalization under GRPO.** Accuracy on the *Numerical* subset of MATH for 0.5B/1.5B/3B models trained with GRPO on (i) MATH L1–L3, (ii) MATH L1–L5, and (iii) GSM8K.

5 DISCUSSION & LIMITATIONS

Interpretation Our results suggest that for Small Language Models (SLMs), the relationship between training data difficulty and alignment gain is not linear, but rather governed by a signal-to-noise ratio. We hypothesize that "hard" problems require multi-hop reasoning chains that exceed the intrinsic representational capacity of sub-3B models. Consequently, during the exploration phase of GRPO, the model rarely generates a correct reasoning trace for these problems, leading to a sparse reward landscape. Instead of providing a directional gradient toward better reasoning, these hard samples likely contribute high-variance gradients that act as noise, potentially destabilizing the learning of simpler concepts. This explains why removing them and focusing on the "learnable" signal from lower-difficulty tiers maintains performance while using $\sim 45\%$ the training steps.

Limitations We acknowledge several constraints in our study. First, we employed LoRA (Low-Rank Adaptation) Hu et al. (2021) rather than full fine-tuning. While our primary experiments rely on parameter-efficient fine-tuning via LoRA for computational tractability, we verified through a full fine-tuning ablation on the 0.5B model (detailed in Appendix F) that the capacity boundary persists even with all parameters unfrozen. This confirms the saturation is an intrinsic property of the model's representational scale rather than an optimization constraint. Second, our study is bounded by the scale of standard academic benchmarks (GSM8K Cobbe et al. (2021), MATH Hendrycks et al. (2021)). The dynamics of difficulty scaling might differ in regimes with massive-scale synthetic data Gunasekar et al. (2023) (e.g., millions of samples), where brute-force memorization could mimic reasoning. Finally, our findings are specific to the 0.5B – 3B parameter regime; these trends likely invert for frontier models (70B+), where capacity is sufficient to leverage complex training signals.

6 CONCLUSION

In this work, we challenge the prevailing assumption that "harder is better" for post-training alignment. We demonstrate that for SLMs in resource-constrained settings, data difficulty is not a strict proxy for alignment quality. Our difficulty-stratified analysis reveals a sharp capacity boundary: models in the 0.5B–3B range saturate on complex reasoning tasks, rendering high-difficulty training samples computationally wasteful. Crucially, we show that training on only the lower-difficulty data matches full-dataset performance, offering a clear path to efficient alignment. Our findings suggest a resource-efficient alignment paradigm: train on what the model can learn from, not on what exceeds its capacity.

REFERENCES

- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL <https://arxiv.org/abs/2110.14168>.
- Muong Ding, Chenghao Deng, Jocelyn Choo, Zichu Wu, Aakriti Agrawal, Avi Schwarzschild, Tianyi Zhou, Tom Goldstein, John Langford, Anima Anandkumar, and Furong Huang. Easy2hard-bench: Standardized difficulty labels for profiling llm performance and generalization, 2025. URL <https://arxiv.org/abs/2409.18433>.
- Zhaolin Gao, Joongwon Kim, Wen Sun, Thorsten Joachims, Sid Wang, Richard Yuanzhe Pang, and Liang Tan. Prompt curriculum learning for efficient llm post-training, 2025. URL <https://arxiv.org/abs/2510.01135>.
- Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. Textbooks are all you need, 2023. URL <https://arxiv.org/abs/2306.11644>.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, Xiaokang Zhang, Xingkai Yu, Yu Wu, Z. F. Wu, Zhibin Gou, Zhihong Shao, Zhuoshu Li, Ziyi Gao, Aixin Liu, Bing Xue, Bingxuan Wang, Bochao Wu, Bei Feng, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Hanwei Xu, Honghui Ding, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jingchang Chen, Jingyang Yuan, Jinhao Tu, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, Kaichao You, Kaige Gao, Kang Guan, Kexin Huang, Kuai Yu, Lean Wang, Lecong Zhang, Liang Zhao, Litong Wang, Liyue Zhang, Lei Xu, Leyi Xia, Mingchuan Zhang, Minghua Zhang, Minghui Tang, Mingxu Zhou, Meng Li, Miaojun Wang, Mingming Li, Ning Tian, Panpan Huang, Peng Zhang, Qiancheng Wang, Qinyu Chen, Qiushi Du, Ruiqi Ge, Ruisong Zhang, Ruizhe Pan, Runji Wang, R. J. Chen, R. L. Jin, Ruyi Chen, Shanghao Lu, Shangyan Zhou, Shanhuang Chen, Shengfeng Ye, Shiyu Wang, Shuiping Yu, Shunfeng Zhou, Shuting Pan, S. S. Li, Shuang Zhou, Shaoqing Wu, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wen Liu, Wenfeng Liang, Wenjun Gao, Wenqin Yu, Wentao Zhang, W. L. Xiao, Wei An, Xiaodong Liu, Xiaohan Wang, Xiaokang Chen, Xiaotao Nie, Xin Cheng, Xin Liu, Xin Xie, Xingchao Liu, Xinyu Yang, Xinyuan Li, Xuecheng Su, Xuheng Lin, X. Q. Li, Xiangyue Jin, Xiaojin Shen, Xiaosha Chen, Xiaowen Sun, Xiaoxiang Wang, Xinnan Song, Xinyi Zhou, Xianzu Wang, Xinxia Shan, Y. K. Li, Y. Q. Wang, Y. X. Wei, Yang Zhang, Yanhong Xu, Yao Li, Yao Zhao, Yaofeng Sun, Yaohui Wang, Yi Yu, Yichao Zhang, Yifan Shi, Yiliang Xiong, Ying He, Yishi Piao, Yisong Wang, Yixuan Tan, Yiyang Ma, Yiyuan Liu, Yongqiang Guo, Yuan Ou, Yudian Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanping Huang, Yaohui Li, Yi Zheng, Yuchen Zhu, Yunxian Ma, Ying Tang, Yukun Zha, Yuting Yan, Z. Z. Ren, Zehui Ren, Zhangli Sha, Zhe Fu, Zhean Xu, Zhenda Xie, Zhengyan Zhang, Zhewen Hao, Zhicheng Ma, Zhigang Yan, Zhiyu Wu, Zihui Gu, Zijia Zhu, Zijun Liu, Zilin Li, Ziwei Xie, Ziyang Song, Zizheng Pan, Zhen Huang, Zhipeng Xu, Zhongyu Zhang, and Zhen Zhang. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645 (8081):633–638, September 2025. ISSN 1476-4687. doi: 10.1038/s41586-025-09422-z. URL <http://dx.doi.org/10.1038/s41586-025-09422-z>.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset, 2021. URL <https://arxiv.org/abs/2103.03874>.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021. URL <https://arxiv.org/abs/2106.09685>.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020. URL <https://arxiv.org/abs/2001.08361>.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, Benjamin Bossan, and Marian Tietz. PEFT: State-of-the-art parameter-efficient fine-tuning methods. <https://github.com/huggingface/peft>, 2022.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback, 2022. URL <https://arxiv.org/abs/2203.02155>.

Shubham Parashar, Shurui Gui, Xiner Li, Hongyi Ling, Sushil Vemuri, Blake Olson, Eric Li, Yu Zhang, James Caverlee, Dileep Kalathil, and Shuiwang Ji. Curriculum reinforcement learning from easy to hard tasks improves llm reasoning, 2025. URL <https://arxiv.org/abs/2506.06632>.

Alethea Power, Yuri Burda, Harri Edwards, Igor Babuschkin, and Vedant Misra. Grokking: Generalization beyond overfitting on small algorithmic datasets, 2022. URL <https://arxiv.org/abs/2201.02177>.

- Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL <https://arxiv.org/abs/2412.15115>.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model, 2024. URL <https://arxiv.org/abs/2305.18290>.
- Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. Deepseekmath: Pushing the limits of mathematical reasoning in open language models, 2024. URL <https://arxiv.org/abs/2402.03300>.
- Yuxuan Tong, Xiwen Zhang, Rui Wang, Ruidong Wu, and Junxian He. DART-math: Difficulty-aware rejection tuning for mathematical problem-solving. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL <https://openreview.net/forum?id=zLU21oQjD5>.
- Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023. URL <https://arxiv.org/abs/2307.09288>.
- Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, Nathan Lambert, Shengyi Huang, Kashif Rasul, and Quentin Gallouédec. TRL: Transformers Reinforcement Learning, 2020. URL <https://github.com/huggingface/trl>.
- Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. Emergent abilities of large language models, 2022. URL <https://arxiv.org/abs/2206.07682>.

A EXTENDED RELATED WORK

Our research sits at the intersection of efficient language modeling, reinforcement learning for mathematical reasoning, and data curriculum design.

A.1 ALIGNMENT FOR REASONING

Standard alignment techniques, such as Reinforcement Learning with Human Feedback (RLHF) Ouyang et al. (2022), rely on a trained reward model and a value function (critic) to guide generation. While PPO has been fundamental in aligning models like LLaMA-2 (Touvron et al., 2023), the memory overhead of maintaining a critic model often equal in size to the policy model is prohibitive for efficient training. Direct Preference Optimization (DPO) (Rafailov et al., 2024) circumvents this by optimizing the policy directly on offline preference pairs. However, for mathematical reasoning, where the solution space is deterministic (binary correctness) but the reasoning path is diverse, online exploration is crucial.

We adopt Group Relative Policy Optimization (GRPO) (Shao et al., 2024), which specifically targets this domain. GRPO samples a group of outputs for a single prompt and computes the baseline for the policy gradient from the group mean rather than a separate value function. This reduces memory usage and naturally incentivizes the model to outperform its own average generation, a dynamic well-suited for the iterative improvement of Chain-of-Thought reasoning.

A.2 THE CAPACITY OF SMALL LANGUAGE MODELS (SLMS)

While scaling laws (Kaplan et al., 2020) have historically driven the development of massive models, recent focus has shifted toward efficiency. The emergence of SLMs, typically under 3B parameters, has demonstrated that high-quality data can offset parameter reduction. The “Phi” series (Gunasekar et al., 2023) validated that models trained on “textbook-quality” synthetic data could rival larger models. However, most SLM evaluations focus on Supervised Fine-Tuning (SFT). Our work extends this inquiry into the post-training phase, specifically investigating whether the reasoning capabilities of these compact models can be further elicited through reinforcement learning without inducing collapse.

A.3 DATA COMPLEXITY AND CURRICULUM LEARNING

The relationship between training data complexity and model performance is a subject of active debate. In mathematical reasoning, recent works argue for “upward” complexity scaling. Ding et al. (2025) and Parashar et al. (2025) propose that training on the frontier of difficulty aids in robust generalization. This aligns with findings in larger models where difficult samples provide the gradients necessary to refine complex decision boundaries.

However, our findings suggest a “Capacity Boundary” for SLMs. This echoes the theoretical underpinnings of “grokking” (Power et al., 2022), where certain reasoning behaviors only materialize past a scale threshold. We provide empirical evidence that for sub-3B models, the “hard” data widely advocated for in LLM literature functions as high-variance noise, leading to the negative transfer phenomena observed in our experiments.

B DETAILED EXPERIMENTAL SETUP

We report full training and evaluation details for both datasets used in this work: Hendrycks MATH and GSM8K. Unless explicitly stated, we keep the pipeline and LoRA configuration identical across datasets, varying only dataset-specific prompting, reward computation, and memory-driven batch sizes.

B.1 SHARED SETUP ACROSS ALL EXPERIMENTS

Models We evaluate Qwen2.5 causal language models at three scales: 0.5B, 1.5B, and 3B parameters.

Training Pipeline All experiments follow a two-stage procedure:

1. **Supervised Fine-Tuning (SFT):** Using LoRA adapters on the target dataset.
2. **GRPO:** Using new LoRA adapters initialized on the SFT-merged checkpoint.

LoRA Configuration All stages use the same LoRA configuration:

- **Rank/Alpha:** $r = 64, \alpha = 64$.
- **Regularization:** Dropout 0.0, Bias none.
- **Target Modules:** q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj.

Precision and Memory We conducted all training on a single A100 40GB GPU using BF16 precision. To optimize memory efficiency and throughput, we employed gradient checkpointing (setting `use_cache=False`) and utilized xFormers' memory-efficient attention kernels.

GRPO Generations Constraint GRPO uses $K = 4$ sampled rollouts per prompt. We choose the per-device batch size and gradient accumulation steps to ensure the effective batch size is strictly divisible by K :

$$(\text{per_device_batch} \times \text{grad_accum}) \pmod{K} \equiv 0.$$

B.2 HENDRYCKS MATH EXPERIMENTS

Dataset We use the EleutherAI Hendrycks MATH dataset, aggregating four subsets: *algebra*, *counting_and_probability*, *geometry*, and *number_theory*. We use the official train/test split, allocating 20% of the training set to SFT and the remaining 80% to GRPO.

Eval set We constructed a numerical subset from the full MATH test set by explicitly selecting samples that require integer or fraction answers. Table 1 provides the detailed sample counts across difficulty levels for both the numerical subset (used for primary evaluation) and the full test set.

Table 1: Distribution of Hendrycks MATH test samples across difficulty levels.

| Dataset | Level 1 | Level 2 | Level 3 | Level 4 | Level 5 | Total |
|-----------|---------|---------|---------|---------|---------|--------------|
| Numerical | 214 | 361 | 407 | 466 | 444 | 1,892 |
| Full | 242 | 476 | 585 | 661 | 716 | 2,680 |

Sequence Lengths Max prompt length is 768 tokens; max completion length is 768 tokens (SFT uses a 1536 token combined budget).

Hyperparameters

- **SFT:** LR 2×10^{-5} , 1 epoch, batch size 8, grad accum 4 (cosine schedule, warmup 0.05).
- **GRPO:** LR 3×10^{-6} , 1 epoch, $\beta = 0.08, \epsilon = 0.2$. Weight decay 0.1, max grad norm 0.1.
- **Sampling:** Temp 0.8, top- p 0.95, top- k 0, `mask_truncated_completions=True`.

GRPO Batch Scaling We vary the per-device batch size by model, as shown in Table 2.

Table 2: GRPO batch scaling for Hendrycks MATH. All runs use $K = 4$ generations.

| Model Size | Per-Device Batch | Grad Accum | Effective Batch |
|------------|------------------|------------|-----------------|
| 0.5B | 20 | 1 | 20 |
| 1.5B | 16 | 1 | 16 |
| 3B | 12 | 1 | 12 |

Prompt Template We use an explicit XML-like structure for answer extraction:

```
Solve the math problem.
Put reasoning between <reasoning> and </reasoning>.
Put ONLY the final answer between <answer> and </answer>.
Do not put any other tags inside <answer>...</answer>.

Problem:
{question}

Assistant:
```

Reward Function The reward is the sum of correctness, formatting, and anti-degeneracy penalties:

$$\mathcal{R} = \mathcal{R}_{\text{correct}} + \mathcal{R}_{\text{format}} + \mathcal{R}_{\text{trunc}} + \mathcal{R}_{\text{short}}$$

Correctness ($\mathcal{R}_{\text{correct}}$): Let $\ell \in \{1, \dots, 5\}$ denote the difficulty level. If a valid answer cannot be extracted, we assign a penalty of -6.0 . Otherwise, we compute equivalence (exact/normalized/SymPy). Rewards are level-weighted:

- **Correct:** $\{3.0, 3.5, 4.5, 6.0, 8.0\}$ for Levels 1–5.
- **Wrong:** $\{-1.2, -1.4, -1.7, -2.0, -2.3\}$ for Levels 1–5.

Formatting ($\mathcal{R}_{\text{format}}$): We add a bonus in range $[-0.20, 0.20]$. The exact structure `<reasoning>...</reasoning><answer>...</answer>` yields $+0.15$, with minor adjustments (± 0.01) for tag counts.

Penalties: $\mathcal{R}_{\text{trunc}} = -1.0$ if the max token budget is hit without a closed answer. $\mathcal{R}_{\text{short}} = -0.2$ if length < 100 tokens.

Evaluation We report greedy decoding accuracy.

B.3 GSM8K EXPERIMENTS

Dataset We use the main GSM8K dataset, splitting training into 30% SFT and 70% GRPO (seed 42).

Sequence Lengths Max prompt/completion length is 512 tokens for both stages.

Hyperparameters

- **SFT:** LR 1×10^{-5} , 1 epoch, batch size 8, grad accum 1 (cosine, warmup 0.03).
- **GRPO:** LR 1×10^{-5} , 2 epochs, $\beta = 0.02$, $\epsilon = 0.2$, `loss_type=dr_grpo`.
- **Sampling:** Temp 1.0, top- p 0.95, top- k 0.

GRPO Batch Scaling Batch sizes are larger relative to MATH, scaled by model size (Table 3).

Table 3: GRPO batch scaling for GSM8K ($K = 4$).

| Model Size | Per-Device Batch | Grad Accum | Effective Batch |
|------------|------------------|------------|-----------------|
| 0.5B | 24 | 2 | 48 |
| 1.5B | 20 | 2 | 40 |
| 3B | 16 | 2 | 32 |

Prompt Template We enforce a single-line final answer:

```

Solve the following math word problem.
Write a short step-by-step solution .
Finish with the final answer on its own line in the exact format:
#### <number>

Question:
{question}

Answer:

```

Reward Function Rewards encourage exact numeric accuracy with continuous relative-error shaping:

$$\mathcal{R} = \mathcal{R}_{\text{base}} + \mathcal{R}_{\text{format}} + \mathcal{R}_{\text{len}}$$

Base ($\mathcal{R}_{\text{base}}$): Let \hat{y} be the prediction and y the gold answer.

- **Parse Fail:** -2.0 (missing delimiter) or -0.75 (numeric parse fail).
- **Correct:** $+3.0$ if relative error $< 10^{-6}$.
- **Incorrect:** Continuous penalty scaled by relative error: $-0.5 - 2.5 \cdot \min(1, \text{rel_err})$.

Formatting ($\mathcal{R}_{\text{format}}$): $+0.25$ if the #### delimiter is present.

Length Penalty (\mathcal{R}_{len}): Soft penalty for verbosity > 220 tokens:

$$\mathcal{R}_{\text{len}} = -\min(1.5, 0.006 \cdot (\text{tokens} - 220))$$

Evaluation We report greedy accuracy.

C FAILURE MODE ANALYSIS

To support the “Reasoning Collapse” hypothesis (Section 4.1), we analyze structural correctness and generation behavior across difficulty levels. All results below are from the **MATH Full**.

C.1 STRUCTURAL COLLAPSE: ANSWER EXTRACTION FAILURES BY DIFFICULTY

We define an **answer extraction failure** as any generation that does not contain a parsable `<answer>...</answer>` span, making the output unscorable by our evaluator. Figure 4 shows that extraction failures rise with difficulty for all model sizes, with a sharp divergence at Level 5.

Key trends:

- **0.5B:** High failure throughout (e.g., **12.9%** at L1 to **20.2%** at L5), indicating weak structural control even at easier levels.
- **1.5B:** Relatively controlled at L1–L4 (3.3–8.1%), but a pronounced **failure spike at L5 (18.1%)**.
- **3B:** More resilient across levels, capping at **13.3%** at L5.

C.2 VERIFYING REASONING EFFORT: AVERAGE GENERATION LENGTH

To rule out “early abandonment,” we track the average completion length (tokens) per difficulty level. Figure 5 shows a strong positive relationship between difficulty and generation length for all models.

Taken together, Figures 4 and 5 suggest that failures at high difficulty are not due to shorter outputs, but rather a loss of *end-of-generation structural control* under heavy reasoning load (most evident for the 1.5B model at Level 5).

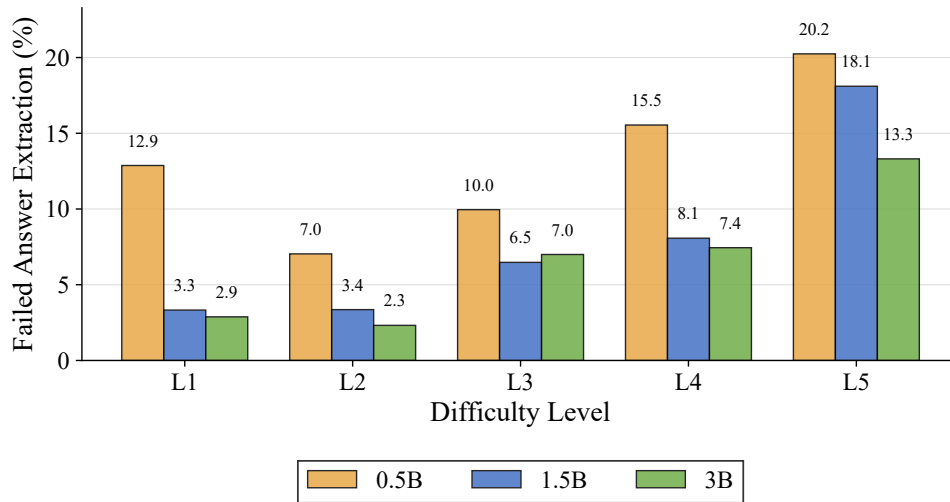


Figure 4: **Answer Extraction Failure Rate vs. Difficulty (MATH Full)**. Percentage of generations without a parsable `<answer>...</answer>` span. Failures increase with difficulty; the 1.5B model spikes at Level 5 (18.1%), while 3B remains more stable (13.3%).

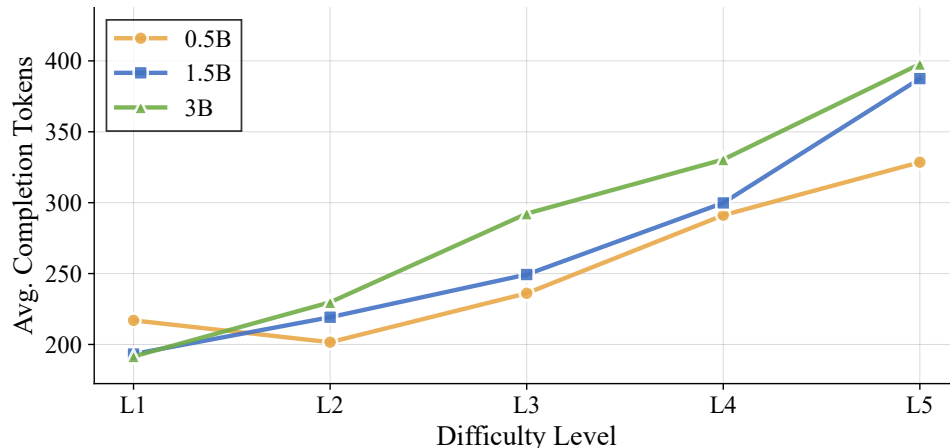


Figure 5: **Average Completion Tokens vs. Difficulty (MATH Full)**. Generation length increases with difficulty across model sizes (approaching ~ 400 tokens at L5 for 1.5B/3B), indicating sustained reasoning effort even when answer extraction fails.

D FINE-GRAINED PERFORMANCE ANALYSIS

In this section, we break down model performance by subject and difficulty level to investigate transfer effects across training distributions. We compare models trained on the full dataset (**MATH**), the Easy subset (**MATH L1-L3**), and the out-of-domain elementary dataset (**GSM8K**).

Summary (Table 4). Subject-wise accuracy on (%) across difficulty Levels L1–L5 for three model sizes (0.5B, 1.5B, 3B), comparing training on **Full MATH (L1–L5)** versus the **Easy subset (MATH L1–L3)**.

Summary (Table 5). Subject-wise accuracy on numerical subset (%) across difficulty Levels L1–L5 for three model sizes (0.5B, 1.5B, 3B), comparing three training sources: **GSM8K**, **Full MATH**, and **MATH L1-L3**.

Table 4: **Impact of Filtering Hard Data.** Accuracy (%) when trained on Full Data (MATH L1–L5) vs. Easy Data (MATH L1–L3).

| Subject | Level | 0.5B | | 1.5B | | 3B | |
|---------------|-------|-------|-------|-------|-------|-------|-------|
| | | L1-L5 | L1-L3 | L1-L5 | L1-L3 | L1-L5 | L1-L3 |
| Algebra | L1 | 51.1 | 57.8 | 83.0 | 78.5 | 86.7 | 87.4 |
| | L2 | 38.8 | 41.8 | 61.7 | 60.2 | 72.6 | 70.6 |
| | L3 | 26.8 | 32.6 | 50.6 | 56.7 | 67.8 | 66.7 |
| | L4 | 18.0 | 20.1 | 44.9 | 45.9 | 51.6 | 63.6 |
| | L5 | 10.1 | 12.7 | 24.8 | 30.6 | 36.5 | 37.5 |
| Count/Prob | L1 | 30.8 | 43.6 | 64.1 | 64.1 | 82.1 | 79.5 |
| | L2 | 30.7 | 32.7 | 44.6 | 40.6 | 64.4 | 55.4 |
| | L3 | 13.0 | 16.0 | 42.0 | 42.0 | 49.0 | 49.0 |
| | L4 | 6.3 | 6.3 | 20.7 | 25.2 | 33.3 | 32.4 |
| | L5 | 2.4 | 3.3 | 8.9 | 12.2 | 14.6 | 12.2 |
| Geometry | L1 | 39.5 | 42.1 | 52.6 | 47.4 | 57.9 | 50.0 |
| | L2 | 35.4 | 29.3 | 43.9 | 42.7 | 48.8 | 43.9 |
| | L3 | 19.6 | 22.5 | 39.2 | 40.2 | 46.1 | 43.1 |
| | L4 | 8.8 | 8.0 | 12.0 | 23.2 | 27.2 | 23.2 |
| | L5 | 0.8 | 1.5 | 5.3 | 4.5 | 7.6 | 6.8 |
| Number Theory | L1 | 13.3 | 43.3 | 63.3 | 70.0 | 76.7 | 66.7 |
| | L2 | 21.7 | 25.0 | 40.2 | 41.3 | 60.9 | 57.6 |
| | L3 | 19.7 | 21.3 | 34.4 | 34.4 | 42.2 | 44.3 |
| | L4 | 5.6 | 7.7 | 21.8 | 21.8 | 34.5 | 33.1 |
| | L5 | 3.2 | 4.5 | 11.0 | 12.3 | 16.9 | 19.5 |

Table 5: **Cross-Domain Transfer Analysis.** Accuracy (%) comparing models trained on GSM8K, MATH (L1–L5), and MATH (L1–L3), evaluated by subject and difficulty.

| Subject | Level | 0.5B | | | 1.5B | | | 3B | | |
|---------------|-------|------|-------|-------|------|-------|-------|------|-------|-------|
| | | GSM | L1-L5 | L1-L3 | GSM | L1-L5 | L1-L3 | GSM | L1-L5 | L1-L3 |
| Algebra | L1 | 61.6 | 53.6 | 56.0 | 91.2 | 81.6 | 80.8 | 93.6 | 88.8 | 84.0 |
| | L2 | 49.1 | 40.9 | 44.0 | 73.6 | 63.5 | 62.3 | 83.6 | 76.1 | 72.3 |
| | L3 | 37.8 | 29.5 | 33.2 | 67.9 | 51.3 | 57.0 | 75.6 | 68.9 | 66.8 |
| | L4 | 21.8 | 19.4 | 18.5 | 52.6 | 38.9 | 41.2 | 62.6 | 54.0 | 64.5 |
| | L5 | 12.1 | 6.1 | 11.6 | 27.3 | 21.7 | 22.2 | 34.3 | 29.3 | 31.3 |
| Count/Prob | L1 | 45.2 | 22.6 | 48.4 | 67.7 | 54.8 | 74.2 | 74.2 | 77.4 | 77.4 |
| | L2 | 24.4 | 37.2 | 25.6 | 44.9 | 50.0 | 38.5 | 56.4 | 70.5 | 60.3 |
| | L3 | 15.0 | 10.0 | 10.0 | 30.0 | 41.7 | 46.7 | 45.0 | 58.3 | 51.7 |
| | L4 | 2.9 | 7.1 | 5.7 | 15.7 | 17.1 | 24.3 | 34.3 | 28.6 | 24.3 |
| | L5 | 7.3 | 3.6 | 5.5 | 9.1 | 9.1 | 5.5 | 14.5 | 12.7 | 7.3 |
| Geometry | L1 | 44.8 | 48.3 | 34.5 | 62.1 | 51.7 | 58.6 | 65.5 | 69.0 | 58.6 |
| | L2 | 40.0 | 33.3 | 37.8 | 55.6 | 53.3 | 51.1 | 62.2 | 62.2 | 60.0 |
| | L3 | 24.0 | 16.0 | 16.0 | 46.0 | 52.0 | 46.0 | 50.0 | 46.0 | 48.0 |
| | L4 | 6.3 | 6.3 | 6.3 | 23.8 | 12.7 | 25.4 | 34.9 | 30.2 | 28.6 |
| | L5 | 0.0 | 0.0 | 2.0 | 2.0 | 6.0 | 4.0 | 12.0 | 10.0 | 10.0 |
| Number Theory | L1 | 37.9 | 10.3 | 48.3 | 72.4 | 69.0 | 58.6 | 79.3 | 79.3 | 65.5 |
| | L2 | 32.9 | 20.3 | 22.8 | 44.3 | 44.3 | 43.0 | 63.3 | 65.8 | 60.8 |
| | L3 | 24.0 | 21.2 | 20.2 | 45.2 | 35.6 | 29.8 | 57.7 | 50.9 | 47.1 |
| | L4 | 6.3 | 4.9 | 7.4 | 22.1 | 23.9 | 19.7 | 35.2 | 36.0 | 41.0 |
| | L5 | 1.4 | 2.1 | 2.1 | 9.9 | 10.6 | 12.1 | 18.4 | 17.0 | 19.1 |

E TRAINING DYNAMICS & STABILITY

To verify the optimization stability of Group Relative Policy Optimization (GRPO) across model scales, we analyze the reward trajectories (Mean) and exploration uncertainty (Standard Deviation) throughout the training process.

E.1 REWARD CURVES: THE CAPACITY GAP

Figure 6 illustrates the average reward per training step.

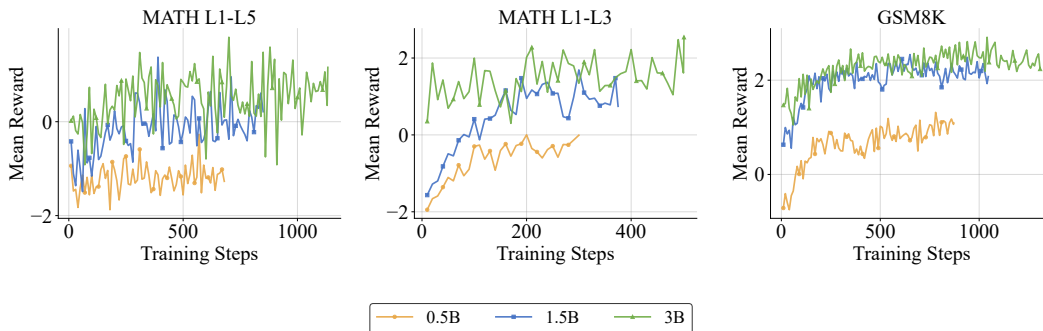


Figure 6: **Reward Mean Trajectories.** **Left/Center:** On the MATH datasets (L1-L5 and L1-L3), the 3B model (Green) demonstrates steady monotonic improvement. In contrast, the 0.5B model (Orange) exhibits a flat or noisy trajectory, indicating an inability to consistently optimize the complex reasoning objective. **Right:** On the GSM8K dataset, all models—including the 0.5B converge reliably, confirming that the instability is specific to the difficulty of the task relative to model capacity.

The trajectories reveal a distinct split in learning behavior:

- **Convergent Learning (GSM8K):** All models, regardless of size, show a healthy upward trend in reward, validating the training hyperparameters.
- **Capacity Stalling (MATH):** On the harder MATH L1-L5 split, the 0.5B model’s reward curve remains near zero or fluctuates randomly. This suggests the model cannot find a gradient direction that consistently leads to correct answers, whereas the 3B model successfully climbs the reward landscape.

E.2 REWARD VARIANCE: GUESSING VS. KNOWING

Figure 7 plots the standard deviation of rewards within each batch, serving as a proxy for model confidence and exploration stability.

- **Healthy Convergence:** On GSM8K, the decreasing standard deviation indicates that the policy is collapsing towards a correct solution mode.
- **High-Variance Exploration:** On MATH L1-L5, the persistent high variance for smaller models implies they are continuously exploring without successfully locking into a high-reward region, effectively “guessing” rather than “learning.”

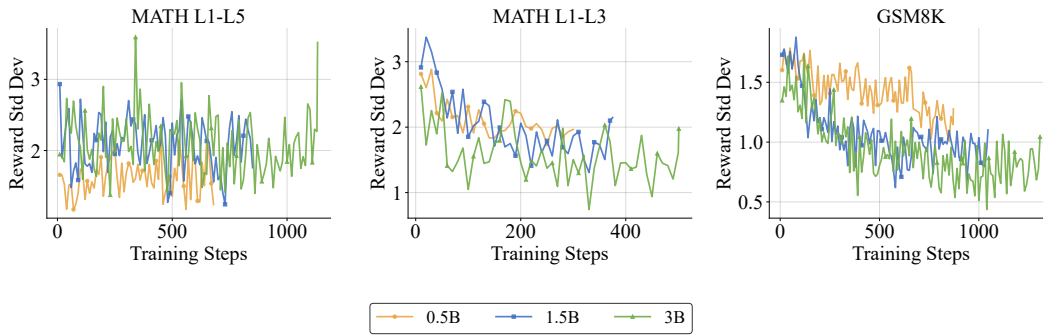


Figure 7: **Reward Standard Deviation:** **Right (GSM8K):** Variance decreases significantly as training progresses (from ~ 1.8 to ~ 0.6), indicating the models are converging on high-confidence solutions. **Left (MATH):** Variance remains high and volatile for the 0.5B and 1.5B models, suggesting they are relying on stochastic exploration (“guessing”) rather than settling on stable reasoning paths.

F FULL FINE-TUNING (FFT) ABLATION

F.1 MOTIVATION

To ensure that the observed capacity boundary is not merely a consequence of the restricted parameter updates inherent to Low-Rank Adaptation (LoRA), we conducted a full fine-tuning (FFT) ablation on the 0.5B model. This experiment tests whether unfrozen parameters allow Small Language Models (SLMs) to internalize complex logic that parameter-efficient methods might bottleneck.

F.2 EXPERIMENTAL SETUP

We replicated the Group Relative Policy Optimization (GRPO) training pipeline using full parameter updates instead of LoRA adapters. We maintained the exact same hyperparameters as the LoRA-based training, with the exception of the batch sizes. Specifically, batch size was set to 8 for both Supervised Fine-Tuning (SFT) and GRPO stages, and the models were trained for 2 epochs. All other reward formulations, sequence lengths, and difficulty stratification protocols remained identical to the primary MATH setup detailed in Appendix B.

F.3 CONCLUSION

As demonstrated in Figure 8, the fully fine-tuned models trained on the complete dataset (L1–L5) achieve similar performance to their counterparts trained exclusively on the lower-difficulty subset (L1–L3). This confirms that high-difficulty problems actively inject noise into the alignment process for SLMs. Because the performance degradation persists even with all parameters unfrozen, we conclude that the capacity boundary is an intrinsic property of the models’ representational scale, not an artifact of the adaptation method.

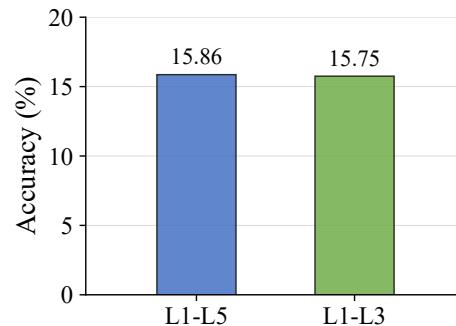


Figure 8: Full Fine-Tuning (FFT) accuracy comparison for 0.5B model on the MATH dataset. Models trained on the easier subset (MATH L1–L3) match the performance of models trained on the full difficulty spectrum (MATH L1–L5), mirroring the LoRA-based capacity boundary.

G CONTROL ANALYSIS (SIZE-MATCHED SUBSAMPLING)

G.1 MOTIVATION

In our primary experiments, the lower-difficulty L1–L3 subset utilizes roughly 45% of the data compared to the full L1–L5 training set. To ensure the observed performance improvements are not simply a byproduct of training on less data, a control experiment was necessary to completely disentangle the effect of problem difficulty from confounding variables such as dataset size and the total number of optimization steps.

G.2 EXPERIMENTAL SETUP

To isolate the variable of difficulty, we constructed a size-matched control dataset. We took a random sample of the full MATH L1–L5 dataset, but capped the total sample count to exactly match the size of the L1–L3 subset. We then trained the 1.5B model on this subset using the identical GRPO hyperparameters and LoRA configuration detailed in Appendix B.

G.3 RESULTS AND ANALYSIS

Remarkably, the size-matched Random 45% subset performed even worse than the full L1–L5 dataset. We hypothesize that this degradation is driven by a deteriorated signal-to-noise ratio. When the model is exposed to the same proportion of “noisy” high-difficulty samples, but is given fewer total optimization steps to lock onto the “clean” learning signal provided by the easier samples, its ability to consolidate stable reasoning patterns is severely compromised.

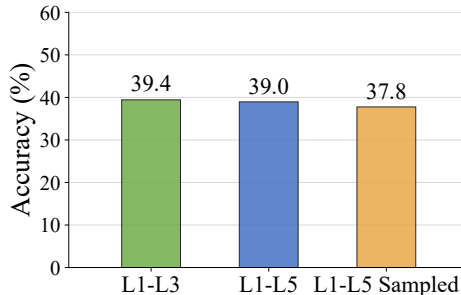


Figure 9: Accuracy comparison of the 1.5B model across three training distributions: the MATH L1–L3 dataset, the MATH L1–L5 subset, and a size-matched random sample of the full dataset MATH L1–L5 (Random 45%).

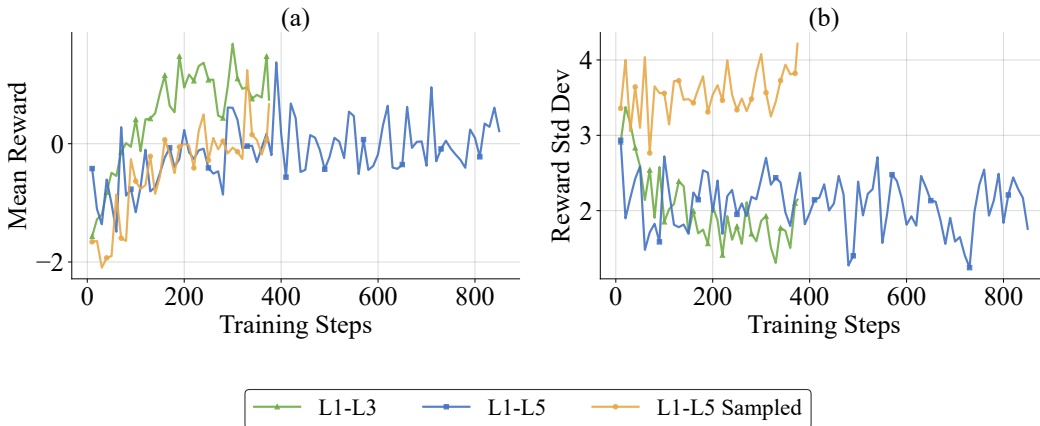


Figure 10: Training dynamics for the size-matched Random 45% control experiment. (a) Mean Reward trajectory. (b) Reward Standard Deviation. The volatile reward and persistently high variance further illustrate the destabilizing effect of hard samples under a reduced step count.

This instability is explicitly reflected in the training dynamics across the three data splits (Figure 10). While the L1-L3 subset establishes a steady upward gradient, the Mean Reward trajectories for both the full L1-L5 and the Sampled L1-L5 struggle to stabilize, with the Sampled L1-L5 showing the most severe degradation. Concurrently, the Mean Standard Deviation for the Sampled L1-L5 remains persistently higher and more volatile

than both the L1-L3 and full L1-L5 runs, indicating that the model is trapped in a state of high-variance exploration without converging on correct reasoning chains.

G.4 CONCLUSION

This confirms that high-difficulty problems actively inject noise into the GRPO optimization process for SLMs, and that our observed gains in Section 4.2 are exclusively due to difficulty stratification, not dataset size.

H RESPONSE ANALYSIS

Setup For each training setting, we include two representative generations: one that is correct and one that is incorrect or unscorable. We also provide some observed patterns in the correct and incorrect or unscorable questions.

Observed success patterns. Analysis of successful generations reveals a compact set of reusable reasoning *primitives* that consistently yield correct results, provided the task lies within the model’s reasoning capabilities:

- **Explicit reduction to canonical forms.** Successful generations efficiently map the problem statement into a standard mathematical representation (e.g., factored inequalities, systems of equations, or modulus reductions), allowing for straightforward algorithmic resolution. (Examples: Sample 9, Sample 13.)
- **Exhaustive case analysis.** When the problem structure requires branching logic (such as absolute values or sign changes), valid solutions systematically enumerate all disjoint cases to ensure complete coverage of the solution space before synthesizing the final result. (Example: Sample 1.)
- **Exploitation of domain invariants.** High-fidelity reasoning avoids redundant derivation by leveraging stable domain properties (e.g., angle bisector ratios, area proportionality, or periodicity) to bypass unnecessary symbolic expansion. (Examples: Sample 5, Sample 13.)
- **Structured constraint modeling.** In multi-constraint scenarios, correct solutions are characterized by the formal declaration of variables and the translation of constraints into a coherent system of equations, facilitating systematic elimination. (Examples: Sample 11, Sample 17.)
- **Definitive termination.** The reasoning process strictly adheres to termination protocols, concluding with a singular, unambiguous output that matches the required format. (Examples: Sample 7, Sample 9.)

Observed failure patterns. Analysis of incorrect or unscorable generations reveals that failures converge on specific recurring modes, consistent with a “capacity boundary” interpretation:

- **Cascading arithmetic divergence.** Minor numerical errors (e.g., in coefficients or discriminants) frequently alter the mathematical regime of the problem (e.g., shifting from real to complex roots). Instead of backtracking, the model often attempts to force a solution through the invalidated path, leading to incoherent results. (Examples: Sample 6, Sample 2, Sample 16.)
- **Inefficient parameterization vs. structural exploitation.** Failures often arise when the model attempts to resolve latent parameters (a, b, c) explicitly, rather than identifying necessary invariants or differences. This misallocation of reasoning resources typically results in algebraic bloat or circular logic without progress. (Example: Sample 2.)
- **Geometric conceptual misalignment.** In geometric reasoning tasks, errors frequently stem from an incorrect mapping of textual descriptions to geometric entities, such as conflating an altitude with a radius or misidentifying axes of rotation. (Examples: Sample 10, Sample 12, Sample 18.)

Gold: 8

Predicted: -

Sample 3: Response of 1.5B model trained on GSM8K (success)

Subject: Algebra (Level 4)**Question:**

Find the constant c such that $(x^2 - 4x + 3)(x + 5) - (x^2 + 4x - 5)(x - c) = 0$ for all x .

Answer:

First, we expand the two factors on the left side of the equation:

$$(x^2 - 4x + 3)(x + 5) = x^3 + 5x^2 - 4x^2 - 20x + 3x + 15 = x^3 + x^2 - 17x + 15$$

$$(x^2 + 4x - 5)(x - c) = x^3 - cx^2 + 4x^2 - 4cx - 5x + 5c = x^3 + (4 - c)x^2 + (-4c - 5)x + 5c$$

Now, we subtract the second factor from the first:

$$x^3 + x^2 - 17x + 15 - (x^3 + (4 - c)x^2 + (-4c - 5)x + 5c) = 0$$

$$x^3 + x^2 - 17x + 15 - x^3 - (4 - c)x^2 - (-4c - 5)x - 5c = 0$$

$$x^2 - 17x + 15 - (4 - c)x^2 + (4c + 5)x - 5c = 0$$

$$(-1 + c)x^2 + (4c + 5 - 17)x + (15 - 5c) = 0$$

$$(-1 + c)x^2 + (4c - 12)x + (15 - 5c) = 0$$

For this equation to be true for all x , the coefficients of x^2 , x , and the constant term must be equal to zero. Therefore, we have the following system of equations:

$$-1 + c = 0$$

$$4c - 12 = 0$$

$$15 - 5c = 0$$

Solving the first equation, we find $c = 1$. Solving the second equation, we find $c = 3$. Solving the third equation, we find $c = 3$. Therefore, the only value of c that satisfies all three equations is

$$c = \boxed{3}.$$

3

Gold: 3

Predicted: 3

Sample 4: Response of 1.5B model trained on GSM8K (failure)

Subject: Algebra (Level 4)**Question:**

What value of x will give the minimum value for $9x^2 + 18x + 7$?

Answer:

To find the minimum value of the quadratic function $9x^2 + 18x + 7$, we can use the formula for the vertex of a parabola, which is given by $x = -\frac{b}{2a}$, where a and b are the coefficients of the quadratic function.

In this case, $a = 9$ and $b = 18$, so we have $x = -\frac{18}{2(9)} = -\frac{18}{18} = -1$.

Therefore, the value of x that will give the minimum value for $9x^2 + 18x + 7$ is $\boxed{-1}$.

1

Gold: -1

Predicted: 1

Sample 5: Response of 3B model trained on GSM8K (success)

Subject: Geometry (Level 5)

Question:

Let $\triangle ABC$ be a triangle such that $AB = 13$, $BC = 14$, and $CA = 15$.
Meanwhile, D is a point on BC such that AD bisects $\angle A$. Find the area of $\triangle ADC$.

Answer:

By the Angle Bisector Theorem, we have $\frac{BD}{DC} = \frac{AB}{AC} = \frac{13}{15}$.
Let $BD = 13x$ and $DC = 15x$. Then $BD + DC = BC = 14$, so $13x + 15x = 14$, which gives $x = \frac{1}{2}$.

Therefore, $BD = 13x = \frac{13}{2}$ and $DC = 15x = \frac{15}{2}$.

By Heron's formula, the area of $\triangle ABC$ is $\sqrt{s(s-AB)(s-BC)(s-CA)}$, where s is the semiperimeter.

The semiperimeter is $s = \frac{AB+BC+CA}{2} = \frac{13+14+15}{2} = 21$.

So the area of $\triangle ABC$ is $\sqrt{21(21-13)(21-14)(21-15)} = \sqrt{21 \cdot 8 \cdot 7 \cdot 6} = 84$.

Since AD bisects $\angle A$, the area of $\triangle ADC$ is $\frac{DC}{BC} \cdot \text{Area of } \triangle ABC = \frac{15}{28} \cdot 84 = 45$.

45

Gold: 45

Predicted: 45

Sample 6: Response of 3B model trained on GSM8K (Failure)

Subject: Algebra (Level 4)

Question:

The equation $y = -6t^2 + 43t$ describes the height (in feet) of a projectile t seconds after it is launched from the surface of Mars at 43 feet per second. In how many seconds will the projectile first reach 77 feet in height? Express your answer as a decimal rounded to the nearest tenth.

Answer:

We need to find the time t when the height y is 77 feet.

So we set $y = 77$ and solve for t :

$$77 = -6t^2 + 43t$$

Rearranging the equation, we get:

$$6t^2 - 43t + 77 = 0$$

We can solve this quadratic equation using the quadratic formula:

$$t = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

where $a = 6$, $b = -43$, and $c = 77$.

Plugging in the values, we get:

$$t = \frac{43 \pm \sqrt{(-43)^2 - 4(6)(77)}}{2(6)}$$

$$t = \frac{43 \pm \sqrt{1849 - 1872}}{12}$$

$$t = \frac{43 \pm \sqrt{1849 - 1872}}{12}$$

$$t = \frac{43 \pm \sqrt{1849 - 1872}}{12}$$

$$t = \frac{43 \pm \sqrt{1849 - 1872}}{12}$$

$$t = \frac{43 \pm \sqrt{1849 - 1872}}{12}$$

$$t = \frac{43 \pm \sqrt{1849 - 1872}}{12}$$

$$t = \frac{43 \pm \sqrt{1849 - 1872}}{12}$$

$$t = \frac{43 \pm \sqrt{1849 - 1872}}{12}$$

$$t = \frac{43 \pm \sqrt{1849 - 1872}}{12}$$

$$t = \frac{43 \pm \sqrt{1849 - 1872}}{12}$$

$$t = \frac{43 \pm \sqrt{1849 - 1872}}{12}$$

$$t = \frac{43 \pm \sqrt{1849 - 1872}}{12}$$

$$t = \frac{43 \pm \sqrt{1849 - 1872}}{12}$$

$$t = \frac{43 \pm \sqrt{1849 - 1872}}{12}$$

$$t = \frac{43 \pm \sqrt{1849 - 1872}}{12}$$

$$t = \frac{43 \pm \sqrt{1849 - 1872}}{12}$$

$$t = \frac{43 \pm \sqrt{1849 - 1872}}{12}$$

$$t = \frac{43 \pm \sqrt{1849 - 1872}}{12}$$

$$t = \frac{43 \pm \sqrt{1849 - 1872}}{12}$$

$$t = \frac{43 \pm \sqrt{1849 - 1872}}{12}$$

Gold: 3.5

Predicted: -

Sample 7: Response of 0.5B model trained on MATH L1-L5 (success)

Subject: Algebra (Level 4)**Question:**If $f(x) = \frac{4x+1}{3}$ what is the value of $[f^{-1}(1)]^{-1}$?**Answer:**

<reasoning>

We want to find the value of $[f^{-1}(1)]^{-1}$, which means we want to find the value of x such that $f(x) = 1$.

We know that $f(x) = \frac{4x+1}{3}$, so we set $\frac{4x+1}{3} = 1$ and solve for x .

Multiplying both sides by 3, we get $4x + 1 = 3$.

Subtracting 1 from both sides, we get $4x = 2$.

Dividing both sides by 4, we get $x = \frac{1}{2}$.

Therefore, $[f^{-1}(1)]^{-1} = \frac{1}{1/2} = \boxed{2}$.

</reasoning>

<answer>2</answer>

Gold: 2

Predicted: 2

Sample 8: Response of 0.5B model trained on MATH L1-L5 (Failure)

Subject: Counting and Probability (Level 4)**Question:**

Our physics club has 20 members, among which we have 3 officers: President, Vice President, and Treasurer. However, one member, Alex, hates another member, Bob. How many ways can we fill the offices if Alex refuses to serve as an officer if Bob is also an officer? (No person is allowed to hold more than one office.)

Answer:

<reasoning>

Let A be the set of members who hate Bob, and B be the set of members who hate Alex. Then $|A| = 15$ and $|B| = 15$. We want to count the number of ways to assign the members of A and B to the three positions, and then subtract the number of ways to assign the members of A and B to the three positions, where the first member of A is assigned to the President's position, the second member of A is assigned to the Vice President's position, and the third member of A is assigned to the Treasurer's position. There are 15 ways to assign the members of A to the three positions, and 15 ways to assign the members of B to the three positions, so there

are $15 \cdot 15 = 225$ ways to assign the members of A and B to the three positions. However, we must also subtract the number of ways to assign the members of A and B to the three positions, where the first member of A is assigned to the President's position, the second member of A is assigned to the Vice President's position, and the third member of A is assigned to the Treasurer's position. There are 15 ways to assign the members of A to the three positions, and 15 ways to assign the members of B to the three positions, so there are $15 \cdot 15 = 225$ ways to assign the members of A and B to the three positions. However, we must also subtract the number of ways to assign the members of A and B to the three positions, where the first member of A is assigned to the President's position, the second member of A is assigned to the Vice President's position, and the third member of A is assigned to the Treasurer's position. There are 15 ways to assign the members of A to the three positions, and 15 ways to assign the members of B to the three positions, so there are $15 \cdot 15 = 225$ ways to assign the members of A and B to the three positions.

</reasoning>
<answer>225</answer>

Gold: 6732

Predicted: 225

Sample 9: Response of 1.5B model trained on MATH L1-L5 (Success)

Subject: Algebra (Level 4)

Question:

Let a be the smallest integer satisfying the inequality $x^2 - 15 < 2x$, and let b be the largest integer satisfying the same inequality. What is $b - a$?

Answer:

<reasoning>

To solve the inequality $x^2 - 15 < 2x$, we first rearrange it into a standard quadratic form:

$$x^2 - 2x - 15 < 0.$$

Next, we factor the quadratic expression:

$$(x - 5)(x + 3) < 0.$$

To determine the intervals where this inequality holds, we find the roots of the equation $(x - 5)(x + 3) = 0$, which are $x = 5$ and $x = -3$.

These roots divide the number line into three intervals: $(-\infty, -3)$, $(-3, 5)$, and $(5, \infty)$. We test a point from each interval in the inequality $(x - 5)(x + 3) < 0$:

- For $x \in (-\infty, -3)$, choose $x = -4$: $(-4 - 5)(-4 + 3) = (-9)(-1) = 9 > 0$.
- For $x \in (-3, 5)$, choose $x = 0$: $(0 - 5)(0 + 3) = (-5)(3) = -15 < 0$.
- For $x \in (5, \infty)$, choose $x = 6$: $(6 - 5)(6 + 3) = (1)(9) = 9 > 0$.

Thus, the inequality $(x - 5)(x + 3) < 0$ holds for $x \in (-3, 5)$. The smallest integer in this interval is $a = -2$.

To find the largest integer b that satisfies the inequality, we note that the largest integer in the interval $(-3, 5)$ is $b = 4$.

Finally, we calculate $b - a$:

$$b - a = 4 - (-2) = 4 + 2 = 6.$$

</reasoning>

<answer>6</answer>

Gold: 6

Predicted: 6

Sample 10: Response of 1.5B model trained on MATH L1-L5 (Failure)

Subject: Geometry (Level 4)

Question:

An equilateral triangle of side 12 centimeters is rotated about an altitude to form a cone. What is the number of cubic centimeters in the volume of the cone?

Answer:

<reasoning>

The altitude of the equilateral triangle is 12 centimeters, so the radius of the cone is 12 centimeters. The height of the cone is also 12 centimeters. The volume of a cone is given by the formula $V = (1/3)\pi r^2 h$, where r is the radius and h is the height. Plugging in the values, we get $V = (1/3)\pi(12^2)(12) = 576\pi$ cubic centimeters.

</reasoning>

<answer>576π </answer >

Gold: $72\pi\sqrt{3}$

Predicted: 576π

Sample 11: Response of 3B model trained on MATH L1-L5 (Success)

Subject: Algebra (Level 5)

Question:

Anna, Bertram, Carli, and David have a competition to see which of them can hold their breath for the longest time period, in minutes. If Bertram, Carli, and David add their times together, the resulting sum is three times the length of time that Anna can hold her breath. Similarly, if Anna, Carli, and David sum their times, the result is four times Bertram's time period, and if Anna, Bertram, and David sum their times, the result is twice Carli's time. Finally, eight times Anna's time plus ten times Bertram's time plus six times Carli's time equals two fifths of an hour. If the length of time that David can hold his breath is expressed in minutes as a simplified fraction, what is the sum of the numerator and the denominator?

Answer:

<reasoning>

Let's denote the times that Anna, Bertram, Carli, and David can hold their breath as A , B , C , and D respectively. We are given the following equations:

$$1. \quad B + C + D = 3A \quad 2. \quad A + C + D = 4B \quad 3. \quad A + B + D = 2C \quad 4.$$

$$8A + 10B + 6C = \frac{2}{5} \times 60 = 24$$

We can solve these equations step by step. First, let's subtract the first equation from the second equation:

$$(A + C + D) - (B + C + D) = 4B - 3A$$

$$A - B = 4B - 3A$$

$$4A = 5B$$

$$B = \frac{4A}{5}$$

Next, let's subtract the first equation from the third equation:

$$(A + B + D) - (B + C + D) = 2C - 3A$$

$$A - C = 2C - 3A$$

$$4A = 3C$$

$$C = \frac{4A}{3}$$

Now, let's substitute $B = \frac{4A}{5}$ and $C = \frac{4A}{3}$ into the fourth equation:

$$8A + 10\left(\frac{4A}{5}\right) + 6\left(\frac{4A}{3}\right) = 24$$

$$8A + 8A + 8A = 24$$

$$24A = 24$$

$$A = 1$$

Now that we have $A = 1$, we can find B , C , and D :

$$B = \frac{4A}{5} = \frac{4 \times 1}{5} = \frac{4}{5}$$

$$C = \frac{4A}{3} = \frac{4 \times 1}{3} = \frac{4}{3}$$

$$D = 3A - B - C = 3 \times 1 - \frac{4}{5} - \frac{4}{3} = 3 - \frac{4}{5} - \frac{4}{3} = \frac{45}{15} - \frac{12}{15} - \frac{20}{15} = \frac{13}{15}$$

The length of time that David can hold his breath is $\frac{13}{15}$ minutes. The sum of the numerator and the denominator is $13 + 15 = 28$.

Therefore, the answer is:

28

</reasoning>

<answer>28</answer>

Gold: 28

Predicted: 28

Sample 12: Response of 3B model trained on MATH L1-L5 (Failure)

Subject: Geometry (Level 5)

Question:

In the diagram, K , O and M are the centers of the three semi-circles. Also, $OC = 32$ and $CB = 36$.

```
[asy]
pair A, K, O, C, M, B, X, Y, Z, J, T;
O=(0,0);
C=(32,0);
M=(50,0);
B=(68,0);
A=(-68,0);
K=(A+C)/2;
X=(0,68);
Y=(-18,50);
Z=(50,18);
J=(7,43.3);
T=(59,15.6);
path nom, bigc, middlec, smallc;
nom=A--B--(100,100)--(-100,100)--cycle;
bigc=A..X..B--cycle;
middlec=A..Y..C--cycle;
smallc=C..Z..B--cycle;
fill(bigc, gray(.5));
fill(middlec, white);
fill(smallc, white);
draw(smallc);
draw(middlec);
draw(bigc);
draw(A--B);
label("A", A, S);
```

```

label("K", K, S);
label("O", O, S);
label("M", M, S);
label("C", C, S);
label("B", B, S);
label("S", J, SW);
label("E", T, SW);
label("l", (.9(J-T)+J), NW);
draw((.9(J-T)+J)--(.5(T-J)+T));
dot(K);
dot(O);
dot(M);
dot(J);
dot(T);
[/asy]

```

Line l is drawn to touch the smaller semi-circles at points S and E so that KS and ME are both perpendicular to l . Determine the area of quadrilateral $KSEM$.

Answer:

<reasoning>

Let r be the radius of the smaller semi-circle. Then $OC = 32$ and $CB = 36$, so $OB = 32 + 36 = 68$. Since K is the center of the smaller semi-circle, $OK = r$. By the Pythagorean Theorem, $r^2 + 32^2 = 68^2$, so $r^2 = 68^2 - 32^2 = 4096 - 1024 = 3072$. Therefore, $r = \sqrt{3072} = 16\sqrt{12} = 32\sqrt{3}$.

Since KS and ME are both perpendicular to l , KS and ME are both radii of the smaller semi-circle. Therefore, $KS = ME = 32\sqrt{3}$.

Since K , O , and M are the centers of the three semi-circles, $KO = OM = 32\sqrt{3}$. Therefore, $KO = OM = 32\sqrt{3}$.

Since K , O , and M are the centers of the three semi-circles, $KO = OM = 32\sqrt{3}$. Therefore, $KO = OM = 32\sqrt{3}$.

Since K , O , and M are the centers of the three semi-circles, $KO = OM = 32\sqrt{3}$. Therefore, $KO = OM = 32\sqrt{3}$.

Since K , O , and M are the centers of the three semi-circles, $KO = OM = 32\sqrt{3}$. Therefore, $KO = OM = 32\sqrt{3}$.

Since K , O , and M are the centers of the three semi-circles, $KO = OM = 32\sqrt{3}$. Therefore, $KO = OM = 32\sqrt{3}$.

Since K , O , and M are the centers of the three semi-circles, $KO = OM = 32\sqrt{3}$. Therefore, $KO = OM = 32\sqrt{3}$.

Since K , O , and M are the centers of the three semi-circles, $KO = OM = 32\sqrt{3}$. Therefore, $KO = OM = 32\sqrt{3}$.

Since K , O , and M are the centers of the three semi-circles, $KO = OM = 32\sqrt{3}$. Therefore, $KO = OM = 32\sqrt{3}$.

Since K , O , and M are the centers of the three semi-circles, $KO = OM = 32\sqrt{3}$. Therefore, $KO = OM = 32\sqrt{3}$.

Since K , O , and M are the centers of the three semi-circles, $KO = OM = 32\sqrt{3}$. Therefore, $KO = OM = 32\sqrt{3}$.

Since K , O , and M are the centers of the three semi-circles, $KO = OM = 32\sqrt{3}$.

Gold: 2040

Predicted: -

Sample 13: Response of 0.5B model trained on MATH L1-L3 (Success)

Subject: Geometry (Level 4)

Question:
Compute $\sin 1755^\circ$.

Answer:
<reasoning>
To compute $\sin 1755^\circ$, we can use the periodicity of the sine function. The sine function has a period of 360° , which means that $\sin x = \sin(x + 360^\circ k)$ for any integer k .

First, we can reduce 1755° to an angle between 0° and 360° by subtracting multiples of 360° :

$$1755^\circ - 5 \cdot 360^\circ = 1755^\circ - 1800^\circ = -45^\circ$$

Since -45° is still not in the range of 0° to 360° , we add 360° to get:

$$-45^\circ + 360^\circ = 315^\circ$$

Now, we can use the periodicity of the sine function to find $\sin 315^\circ$:

$$\sin 315^\circ = \sin(360^\circ - 45^\circ) = \sin(-45^\circ) = -\sin 45^\circ$$

Since $\sin 45^\circ = \frac{\sqrt{2}}{2}$, we have:

$$\sin 315^\circ = -\frac{\sqrt{2}}{2}$$

Therefore, $\sin 1755^\circ = -\frac{\sqrt{2}}{2}$.
</reasoning>
<answer> $-\frac{\sqrt{2}}{2}$ </answer>

Gold: $-\frac{\sqrt{2}}{2}$ **Predicted:** $-\frac{\sqrt{2}}{2}$

Sample 14: Response of 0.5B model trained on MATH L1-L3 (Failure)

Subject: Algebra (Level 5)

Question:
Simplify $\frac{3}{\sqrt[5]{16}} + \frac{1}{\sqrt{3}}$ and rationalize the denominator. The result can be expressed in the form $\frac{a^2 \sqrt[5]{b} + b\sqrt{a}}{ab}$, where a and b are integers. What is the value of the sum $a + b$?

Answer:
<reasoning>
To simplify the expression, we can combine the fractions over a common denominator:

$$\frac{3}{\sqrt[5]{16}} + \frac{1}{\sqrt{3}} = \frac{3\sqrt[5]{16} + 1\sqrt{3}}{\sqrt[5]{16}\sqrt{3}} = \frac{3\sqrt[5]{16} + 1\sqrt{3}}{\sqrt[5]{48}}$$

Next, we rationalize the denominator by multiplying the numerator and denominator by $\sqrt[5]{48}$:

$$\frac{3\sqrt[5]{16} + 1\sqrt{3}}{\sqrt[5]{48}} \cdot \frac{\sqrt[5]{48}}{\sqrt[5]{48}} = \frac{(3\sqrt[5]{16} + 1\sqrt{3})\sqrt[5]{48}}{48} = \frac{3\sqrt[5]{768} + 1\sqrt{144}}{48}$$

We can further simplify the expression by combining the terms in the numerator:

$$\begin{aligned} \frac{3\sqrt[5]{768} + 1\sqrt{144}}{48} &= \frac{3\sqrt[5]{768} + 12\sqrt{12}}{48} = \frac{3\sqrt[5]{768} + 12\sqrt{12}}{48} \\ &= \frac{3\sqrt[5]{768} + 12\sqrt{12}}{48} = \frac{3\sqrt[5]{768} + 12\sqrt{12}}{48} \\ &= \frac{3\sqrt[5]{768} + 12\sqrt{12}}{48} = \frac{3\sqrt[5]{768} + 12\sqrt{12}}{48} \\ &= \frac{3\sqrt[5]{768} + 12\sqrt{12}}{48} = \frac{3\sqrt[5]{768} + 12\sqrt{12}}{48} \\ &= \frac{3\sqrt[5]{768} + 12\sqrt{12}}{48} = \frac{3\sqrt[5]{768} + 12\sqrt{12}}{48} \\ &= \frac{3\sqrt[5]{768} + 12\sqrt{12}}{48} = \frac{3\sqrt[5]{768} + 12\sqrt{12}}{48} \\ &= \frac{3\sqrt[5]{768} + 12\sqrt{12}}{48} = \frac{3\sqrt[5]{768} + 12\sqrt{12}}{48} \\ &= \frac{3\sqrt[5]{768} + 12\sqrt{12}}{48} = \frac{3\sqrt[5]{768} + 12\sqrt{12}}{48} = \end{aligned}$$

Gold: 5

Predicted: -

Sample 15: Response of 1.5B model trained on MATH L1-L3 (Success)

Subject: Number Theory (Level 4)

Question:

How many of the same digits are found in the base 7 and base 8 representations of 629_{10} ? For example, 121_3 and 413_5 would have one digit in common.

Answer:

<reasoning>

To find the number of common digits between the base 7 and base 8 representations of 629_{10} , we first need to convert 629_{10} to its base 7 and base 8 equivalents.

First, let's convert 629_{10} to base 7:

1. Divide 629 by 7: $629 \div 7 = 89$ remainder 6.
2. Divide 89 by 7: $89 \div 7 = 12$ remainder 5.
3. Divide 12 by 7: $12 \div 7 = 1$ remainder 5.
4. Divide 1 by 7: $1 \div 7 = 0$ remainder 1.

Reading the remainders from bottom to top, we get $629_{10} = 1556_7$.

Next, let's convert 629_{10} to base 8:

1. Divide 629 by 8: $629 \div 8 = 78$ remainder 5.
2. Divide 78 by 8: $78 \div 8 = 9$ remainder 6.
3. Divide 9 by 8: $9 \div 8 = 1$ remainder 1.
4. Divide 1 by 8: $1 \div 8 = 0$ remainder 1.

Reading the remainders from bottom to top, we get $629_{10} = 1165_8$.

Now, we compare the digits of 1556_7 and 1165_8 :

- The digit '1' appears in both numbers.
- The digit '5' appears in both numbers.
- The digit '6' appears in both numbers.

Thus, there are 3 common digits between the base 7 and base 8 representations of 629_{10} .

Therefore, the final answer is:

<answer>3</answer>

</reasoning>

Gold: 3

Predicted: 3

Sample 16: Response of 1.5B model trained on MATH L1-L3 (Failure)

Subject: Geometry (Level 4)

Question:

If the degree measure of an arc of a circle is increased by 20% and the radius of the circle is increased by 25%, by what percent does the length of the arc increase?

Answer:

<reasoning>

Let's denote the original degree measure of the arc as θ and the original radius of the circle as r . The length of the arc L is given by the formula:

$$L = \frac{\theta}{360} \times 2\pi r$$

If the degree measure of the arc is increased by 20%, the new degree measure θ' is:

$$\theta' = \theta + 0.2\theta = 1.2\theta$$

If the radius of the circle is increased by 25%, the new radius r' is:

$$r' = r + 0.25r = 1.25r$$

The new length of the arc L' is:

$$\begin{aligned} L' &= \frac{\theta'}{360} \times 2\pi r' = \frac{1.2\theta}{360} \times 2\pi \times 1.25r \\ &= \frac{1.2\theta \times 2\pi \times 1.25r}{360} \\ &= \frac{1.2 \times 2 \times 1.25 \times \theta \times r}{360} \\ &= \frac{3 \times 1.25 \times \theta \times r}{360} = \frac{3.75 \times \theta \times r}{360} \\ &= \frac{3.75}{360} \times \theta \times r = \frac{3.75}{360} \times \theta \times r \\ &= \frac{3.75}{360} \times \theta \times r = \frac{3.75}{360} \times \theta \times r \\ &= \frac{3.75}{360} \times \theta \times r = \frac{3.75}{360} \times \theta \times r \\ &= \frac{3.75}{360} \times \theta \times r = \frac{3.75}{360} \times \theta \times r \\ &= \frac{3.75}{360} \times \theta \times r = \frac{3.75}{360} \times \theta \times r \\ &= \frac{3.75}{360} \times \theta \times r = \frac{3.75}{360} \times \theta \times r \\ &= \frac{3.75}{360} \times \theta \times r = \frac{3.75}{360} \times \theta \times r \\ &= \frac{3.75}{360} \times \theta \times r = \frac{3.75}{360} \times \theta \times r \end{aligned}$$

$$C = 1$$

Therefore, Camille is 1 year old.
</reasoning>

<answer>1</answer>

Gold: 1

Predicted: 1

Sample 18: Response of 3B model trained on MATH L1-L3 (Failure)

Subject: Geometry (Level 5)

Question:

Rhombus $ABCD$, with side length 6, is rolled to form a cylinder of volume 6 by taping \overline{AB} to \overline{DC} . What is $\sin(\angle ABC)$?

Answer:

<reasoning>

To solve this problem, we need to find the sine of the angle between the sides of the rhombus that are taped together to form the cylinder.

First, let's find the height of the cylinder. Since the volume of the cylinder is given as 6, we can use the formula for the volume of a cylinder, which is $V = \pi r^2 h$, where V is the volume, r is the radius, and h is the height.

We know that the volume is 6, so we can set up the equation:

$$6 = \pi r^2 h$$

Since the side length of the rhombus is 6, we can use the Pythagorean theorem to find the height of the rhombus. The height of the rhombus is the same as the height of the cylinder, so we can set up the equation:

$$h = \sqrt{6^2 - r^2}$$

Now we can substitute this expression for h into the volume equation:

$$6 = \pi r^2 \sqrt{6^2 - r^2}$$

To solve for r , we can square both sides of the equation:

$$36 = \pi r^2 (6^2 - r^2)$$

Now we can solve for r^2 :

$$r^2 = 36 / (\pi(6^2 - r^2))$$

We can simplify this equation by dividing both sides by 36:

$$r^2 / 36 = 1 / (\pi(6^2 - r^2))$$

Now we can solve for r^2 : $r^2 = 36 / (\pi(36 - r^2))$

Now we can solve for r^2 :

$$r^2 = 36 / (36\pi - \pi r^2)$$

Now we can solve for r^2 :

$$r^2 = 36 / (36\pi - \pi r^2)$$

Now we can solve for r^2 :

$$r^2 = 36 / (36\pi - \pi r^2)$$

Now we can solve for r^2 :

$$r^2 = 36 / (36\pi - \pi r^2)$$

