Efficient Reinforcement Finetuning via Adaptive Curriculum Learning

Taiwei Shi[†], Yiyang Wu[†], Linxin Song[†], Tianyi Zhou[▽], Jieyu Zhao[†]
[†]University of Southern California, [▽]University of Maryland, College Park
{taiweish, wuwendy, linxinso, jieyuz}@usc.edu, tianyi@umd.edu

Code: github.com/limenlp/verl
Dataset: huggingface.co/datasets/lime-nlp/DeepScaleR_Difficulty

Abstract

Reinforcement finetuning (RFT) has shown great potential for enhancing the mathematical reasoning capabilities of large language models (LLMs), but it is often sample- and compute-inefficient, requiring extensive training. In this work, we introduce ADARFT (*Adaptive Curriculum Reinforcement Finetuning*), a method that significantly improves both the efficiency and final accuracy of RFT through adaptive curriculum learning. ADARFT dynamically adjusts the difficulty of training problems based on the model's recent reward signals, ensuring that the model consistently trains on tasks that are challenging but solvable. This adaptive sampling strategy accelerates learning by maintaining an optimal difficulty range, avoiding wasted computation on problems that are too easy or too hard. ADARFT requires only a lightweight extension to standard RFT algorithms like Proximal Policy Optimization (PPO), without modifying the reward function or model architecture. Experiments on competition-level math datasets—including AMC, AIME, and IMO-style problems—demonstrate that ADARFT significantly improves both training efficiency and reasoning performance. We evaluate ADARFT across multiple data distributions and model sizes, showing that it reduces training time by up to 2$\times$ and improves accuracy by a considerable margin, offering a more scalable and effective RFT framework.

1 Introduction

Reinforcement Finetuning (RFT) has proven effective for improving large language models (LLMs) with task-specific goals (DeepSeek-AI et al., 2025; OpenAI et al., 2024b). By optimizing policies with reward signals, RFT surpasses supervised finetuning (SFT) in targeted learning. However, RFT remains sample-inefficient and computationally costly due to repeated rollouts, reward computation, and policy updates (Ahmadian et al., 2024; Kazemnejad et al., 2024; Li et al., 2024; Hu, 2025; Cui et al., 2025). Recent work has sought to improve efficiency through algorithmic simplifications (e.g., RAFT (Dong et al., 2023), GRPO (DeepSeek-AI et al., 2025), ReMax (Li et al., 2024)) or data-centric strategies (e.g., LIMO (Ye et al., 2025), LIMR (Li et al., 2025)). While effective, these often trade off stability, rely on static filtering, or require heavy model-specific processing. Curriculum-style methods (Wen et al., 2025; Luo et al., 2025; Song et al., 2025) and online filtering (Bae et al., 2025; Yu et al., 2025) offer partial adaptivity but remain coarse-grained or expensive. We introduce ADARFT, a lightweight, adaptive curriculum method that dynamically aligns training difficulty with model ability. By maintaining a target difficulty updated from recent rewards, ADARFT samples problems that are neither trivial nor intractable, enabling steady progression. Unlike staged curricula or rollout-heavy filtering, ADARFT is general, model-agnostic, and easily integrated into standard

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Efficient Reasoning.

RL methods such as Proximal Policy Optimization (PPO) (Schulman et al., 2017b). Experiments on competition-level math datasets (AMC, AIME, IMO-style) show that ADARFT improves both training efficiency and final accuracy, achieving up to $2\times$ faster convergence.

2 ADARFT

We aim to improve the performance of a policy model π_θ for solving mathematical problems through adaptive curriculum learning. Finetuning on problems that are too easy or too hard leads to poor learning outcomes. Instead, the model should be trained on problems whose difficulty is close to the model's current capability. We frame this as an adaptive curriculum learning problem and propose ADARFT, which adaptively adjusts the target difficulty to keep training problems within a suitable difficulty range. ADARFT is compatible with a variety of RL algorithms (e.g, GRPO, PPO); in this work, we instantiate it with PPO and refer to this variant as ADARFT (PPO). Let D be a dataset of mathematical problems, each annotated with a precomputed difficulty score d_i. The score can be either human-annotated or model-estimated. The objective is to train a policy π_θ that improves its problem-solving ability by dynamically adjusting the training curriculum according to the model's current performance. Our proposed algorithm, ADARFT, is shown in Algorithm 1. The core idea is simple: at each iteration, the model is trained on problems whose difficulty is closest to a target level T. After updating the policy with rewards from these problems, T is shifted upward or downward depending on the model's success rate. High performance increases T (harder tasks), while low performance decreases it (easier tasks). This creates a self-adjusting curriculum that avoids both stagnation on easy problems and frustration on unsolvable ones. The implementation details and the theoretical justifications can be found in Appendix B and D.1.

Algorithm 1 ADARFT – Adaptive Curriculum Reinforcement Finetuning

Input: Data source D with difficulty scores $\{d_i\}$, policy model π_θ, reward function $R(\cdot,\cdot)$, batch size B, initial target difficulty T, step size η, sensitivity α, target reward β, difficulty bounds $d_{\min}, d_{\max}$
Select RL algorithm $\mathcal{A}$ (e.g., PPO, GRPO, REINFORCE++)
while training is not finished **do**
  Compute absolute differences from target difficulty: $\Delta_i = |d_i - T| \quad \forall i \in \{1,\dots,|D|\}$
  Sort and select top B samples closest to target difficulty: $X \leftarrow \{s_1, s_2, \dots, s_B\}$
  Generate responses using policy model: $G = \pi_\theta(X)$
  Compute average reward: $R_{avg} \leftarrow \frac{1}{|X|}\sum_{i=1}^{|X|} R(X_i, G_i)$
  Update policy: $\pi_\theta \leftarrow \mathcal{A}(\pi_\theta, X, G, R)$
  Update and clip target difficulty: $T' \leftarrow \text{clip}(T + \eta \cdot \tanh(\alpha \cdot (R_{avg} - \beta)), d_{\min}, d_{\max})$
  Update sampler: $T \leftarrow T'$
end while

3 Experiments

3.1 Difficulty Estimation

Accurate estimation of problem difficulty is critical for ADARFT. For difficulty estimation, we select the Qwen 2.5 MATH 7B model (Qwen et al., 2025) because it demonstrates a balanced solving ability. For each problem, the difficulty score is computed as: $d_i = 100 \times \left(1 - \frac{\text{successful attempts on problem } i}{n}\right)$, where n is the number of attempts per problem. In our setup, we use $n = 128$. We evaluated the stability and reliability of our difficulty estimates and found that they correlated well with human performance (see Appendix C).

3.2 Dataset

We use the DeepScaleR dataset (Luo et al., 2025) as the training set. In practice, we do not have control over the exact difficulty distribution of the data collected for training. This motivates our investigation into how different difficulty distributions influence ADARFT. To this end, we construct three distinct distributions from the DeepScaleR dataset. The first is a `skew-difficult` distribution, where most problems are challenging. The second is a `skew-easy` distribution, where most problems are relatively easy. The third is a `uniform` distribution, where problems are evenly balanced across all difficulty levels. Each of these three distributions includes 10,000 samples. For evaluation, we use six benchmark datasets to assess the model's performance across different levels of difficulty and mathematical reasoning. The first benchmark, MATH 500 (Lightman et al., 2023), GSM8K (Cobbe et al., 2021), OlympiadBench (He et al., 2024), Minerva Math (Lewkowycz et al., 2022), AMC 23, and AIME 24.

3.3 Training Setup

We trained two models on the three difficulty-based distributions of the DeepScaleR dataset described in Section 3.2: Qwen 2.5 7B and Qwen 2.5 MATH 1.5B. This setup allows us to evaluate the effectiveness of ADARFT on models with different initial performance levels when exposed to skew-difficult, skew-easy, and uniform problem distributions. All models were trained using three different approaches: (1) the standard PPO algorithm, (2) ADARFT (PPO), our method that integrates adaptive curriculum learning with PPO (see Section 2), and (3) PPO with filtered data, a strong baseline that trains PPO using filtered data based on pass@k accuracy. For the data filtering baseline (3), we remove examples that are either too easy or too hard, excluding problems with solved percentages $\leq 10\%$ or $\geq 90\%$. The implementation details can be found in Appendix E.

4 Results and Analysis

As shown in Figure 1 and Table 1, models trained with ADARFT consistently require fewer training steps to match the performance of those trained with standard PPO or PPO on filtered data. In addition to improved sample efficiency, ADARFT also achieves faster average training time per step across nearly all settings, as reported in Table 1. This is largely due to the fact that easier problems require fewer tokens to solve. As a result, curriculum learning's tendency to prioritize shorter, easier problems early in training leads to shorter sequences on average, reducing per-step compute and improving overall training throughput.

In addition to improving efficiency, ADARFT (PPO) also improves the final model performance. As shown in Table 2, at the end of training (step 100), ADARFT yields consistent improvements in final accuracy across all configurations. The reported averages reflect accuracy across six diverse benchmarks: GSM8K, MATH 500, OlympiadBench, Minerva Math, AMC 23, and AIME 24. On average, models trained with ADARFT (PPO) outperform their PPO-only counterparts in both final accuracy and training efficiency. This improvement is particularly notable in non-uniform data distributions, where curriculum adaptation is most beneficial.

Our findings show that curriculum learning provides the greatest benefits under two key conditions: (1) imbalanced training distributions, and (2) limited model capacity. In skewed distributions, particularly the skew-difficult settings, standard PPO often struggles to gain traction early in training due to insufficient reward signals. ADARFT mitigates this by initially sampling easier problems, enabling the model to bootstrap capabilities before tackling harder content. Conversely, the benefits of ADARFT are less pronounced when the model is strong enough or the data is already well-balanced. In both cases, the model is either already exposed to a representative distribution of task difficulties or finds most problems chal-

Table 1: Average time per step (in seconds) at step 100 and extra steps required to match ADARFT's performance at step 60 (for Qwen 2.5 Math 1.5B) or step 40 (for Qwen 2.5 7B), across different setups and methods.

Model	Setup	Method	Avg Step Time (s)	Extra Steps (%)
Qwen2.5 Math 1.5B	skew-difficult	ADARFT	**122.24**	**+0 (0.0%)**
		PPO	132.95	+43 (71.7%)
		PPO (w/ Filter)	128.20	+49 (81.7%)
	uniform	ADARFT	**121.31**	**+0 (0.0%)**
		PPO	126.82	+34 (56.7%)
		PPO (w/ Filter)	126.35	+52 (86.7%)
	skew-easy	ADARFT	120.52	**+0 (0.0%)**
		PPO	121.15	+16 (26.7%)
		PPO (w/ Filter)	**115.12**	+21 (35.0%)
Qwen2.5 7B	skew-difficult	ADARFT	**239.92**	**+0 (0.0%)**
		PPO	246.21	+24 (60.0%)
		PPO (w/ Filter)	254.22	+25 (62.5%)
	uniform	ADARFT	**234.16**	**+0 (0.0%)**
		PPO	243.82	+13 (32.5%)
		PPO (w/ Filter)	263.11	+23 (57.5%)
	skew-easy	ADARFT	247.44	**+0 (0.0%)**
		PPO	235.27	+20 (50.0%)
		PPO (w/ Filter)	**233.13**	+17 (42.5%)

lenging enough, thus reducing the need for dynamic difficulty adjustment. In addition, we conducted further experiments detailed in Appendix D, including evaluations on datasets with more extreme difficulty distributions (D.2), ablation studies on different target reward β (D.3), difficulty estimation using an LLM-based judge (D.4), and instantiations of ADARFT with alternative RL algorithms (GRPO, REINFORCE++) (D.5). Across all these settings, ADARFT consistently demonstrates effectiveness, highlighting its robustness to diverse data distributions, compatibility with various RL algorithms, and flexibility with different difficulty metrics.

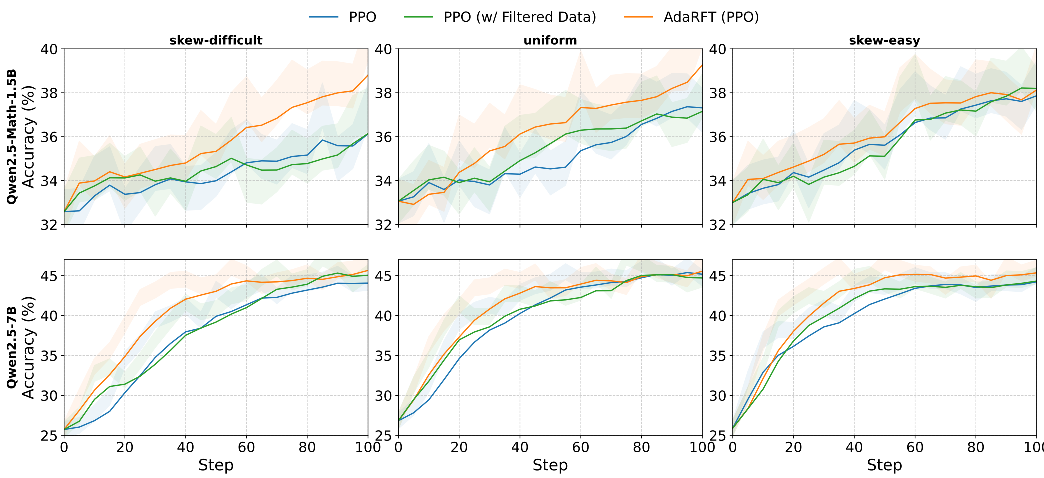


Figure 1: Performance comparison of PPO, PPO with filtered data, and ADARFT (PPO) across different setups (uniform, skew-easy, skew-difficult). Accuracy is the average of MATH 500, GSM8K, AIME 24, AMC 23, OlympiadBench, and Minerva Math. Compared with baselines, ADARFT improves both the accuracy and training efficiency. For clarity, curves are exponentially smoothed.

Table 2: Accuracy (%) at step 100 for every model, setup, and benchmark. ADARFT in this table refers to ADARFT instantiated with PPO, i.e., ADARFT (PPO).

Model	Setup	Method	GSM8K	MATH 500	Olympiad Bench	Minerva Math	AMC 23 (Avg@8)	AIME 24 (Avg@8)	Average
Qwen 2.5 Math 1.5B	skew-difficult	PPO	69.67	64.60	20.65	12.87	47.50	9.17	37.41
		PPO (w/ Filter)	71.65	62.40	20.06	**15.07**	45.00	9.17	37.22
		ADARFT	**74.00**	**66.40**	**20.36**	**15.07**	**55.00**	**12.08**	**40.48**
	uniform	PPO	71.95	65.20	21.10	**15.81**	42.50	6.67	37.20
		PPO (w/ Filter)	72.63	65.80	20.21	13.60	45.00	10.00	37.87
		ADARFT	**74.53**	**66.20**	**21.99**	14.34	**57.50**	**12.08**	**41.11**
	skew-easy	PPO	72.71	**67.40**	19.17	**13.97**	45.00	**12.50**	38.46
		PPO (w/ Filter)	**74.75**	65.20	**20.36**	13.60	45.00	10.00	38.15
		ADARFT	73.62	66.20	19.91	**13.97**	**55.00**	9.17	**39.18**
Qwen 2.5 7B	skew-difficult	PPO	89.69	71.20	23.33	23.53	50.00	11.25	44.17
		PPO (w/ Filter)	88.48	**72.20**	24.37	**25.00**	50.00	12.08	45.35
		ADARFT	**90.98**	71.40	**25.85**	22.43	**52.50**	**15.83**	**46.83**
	uniform	PPO	89.31	72.40	23.63	**25.37**	42.50	**15.00**	44.70
		PPO (w/ Filter)	89.08	**74.40**	23.18	22.43	45.00	13.33	44.57
		ADARFT	**90.14**	72.60	**24.96**	24.26	**55.00**	14.58	**46.92**
	skew-easy	PPO	89.39	**73.60**	23.33	**24.26**	47.50	13.33	45.07
		PPO (w/ Filter)	89.31	71.60	24.22	23.90	47.50	13.33	44.98
		ADARFT	**90.14**	72.60	**25.56**	23.16	**50.00**	**14.17**	**45.94**

5 Conclusion

We propose ADARFT, an adaptive curriculum learning strategy for reinforcement finetuning (RFT) that dynamically matches problem difficulty to a model's evolving skill level. By adjusting a target difficulty based on reward feedback, ADARFT improves both sample and compute efficiency without modifying the reward function or underlying RL algorithm. Experiments across multiple data regimes and model sizes show consistent gains in convergence speed and final accuracy, especially in imbalanced training distributions. This lightweight, scalable approach highlights the value of curriculum-aware training for efficient and robust alignment in structured reasoning tasks.

References

Arash Ahmadian, Chris Cremer, Matthias Gallé, Marzieh Fadaee, Julia Kreutzer, Olivier Pietquin, Ahmet Üstün, and Sara Hooker. Back to basics: Revisiting reinforce style optimization for learning from human feedback in llms, 2024. URL https://arxiv.org/abs/2402.14740.

Sanghwan Bae, Jiwoo Hong, Min Young Lee, Hanbyul Kim, JeongYeon Nam, and Donghyun Kwak. Online difficulty filtering for reasoning oriented reinforcement learning. *arXiv preprint arXiv:2504.03380*, 2025.

Yoshua Bengio, Jérôme Louradour, Ronan Collobert, and Jason Weston. Curriculum learning. In *Proceedings of the 26th Annual International Conference on Machine Learning*, ICML '09, pp. 41–48, New York, NY, USA, 2009. Association for Computing Machinery. ISBN 9781605585161. doi: 10.1145/1553374.1553380. URL https://doi.org/10.1145/1553374.1553380.

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.

Ganqu Cui, Lifan Yuan, Zefan Wang, Hanbin Wang, Wendi Li, Bingxiang He, Yuchen Fan, Tianyu Yu, Qixin Xu, Weize Chen, Jiarui Yuan, Huayu Chen, Kaiyan Zhang, Xingtai Lv, Shuo Wang, Yuan Yao, Xu Han, Hao Peng, Yu Cheng, Zhiyuan Liu, Maosong Sun, Bowen Zhou, and Ning Ding. Process reinforcement through implicit rewards, 2025. URL https://arxiv.org/abs/2502.01456.

Wojciech Marian Czarnecki, Siddhant M. Jayakumar, Max Jaderberg, Leonard Hasenclever, Yee Whye Teh, Simon Osindero, Nicolas Heess, and Razvan Pascanu. Mix&match - agent curricula for reinforcement learning, 2018. URL https://arxiv.org/abs/1806.01780.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, 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, Chenyu Zhang, Chong Ruan, Damai Dai, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, Fuli Luo, Guangbo Hao, Guanting Chen, Guowei Li, H. Zhang, Han Bao, Hanwei Xu, Haocheng Wang, Honghui Ding, Huajian Xin, Huazuo Gao, Hui Qu, Hui Li, Jianzhong Guo, Jiashi Li, Jiawei Wang, Jingchang Chen, Jingyang Yuan, Junjie Qiu, Junlong Li, J. L. Cai, Jiaqi Ni, Jian Liang, Jin Chen, Kai Dong, Kai Hu, 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, 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, Shengfeng Ye, Tao Yun, Tian Pei, Tianyu Sun, T. Wang, Wangding Zeng, Wanjia Zhao, 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, Yuduan Wang, Yue Gong, Yuheng Zou, Yujia He, Yunfan Xiong, Yuxiang Luo, Yuxiang You, Yuxuan Liu, Yuyang Zhou, Y. X. Zhu, Yanhong Xu, 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: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Hanze Dong, Wei Xiong, Deepanshu Goyal, Yihan Zhang, Winnie Chow, Rui Pan, Shizhe Diao, Jipeng Zhang, Kashun Shum, and Tong Zhang. Raft: Reward ranked finetuning for generative foundation model alignment, 2023. URL https://arxiv.org/abs/2304.06767.

Carlos Florensa, David Held, Xinyang Geng, and Pieter Abbeel. Automatic goal generation for reinforcement learning agents, 2018. URL https://arxiv.org/abs/1705.06366.

Dongyoung Go, Tomasz Korbak, Germán Kruszewski, Jos Rozen, Nahyeon Ryu, and Marc Dymetman. Aligning language models with preferences through f-divergence minimization. *arXiv preprint arXiv:2302.08215*, 2023.

Tuomas Haarnoja, Haoran Tang, Pieter Abbeel, and Sergey Levine. Reinforcement learning with deep energy-based policies. In *International conference on machine learning*, pp. 1352–1361. PMLR, 2017.

Chaoqun He, Renjie Luo, Yuzhuo Bai, Shengding Hu, Zhen Leng Thai, Junhao Shen, Jinyi Hu, Xu Han, Yujie Huang, Yuxiang Zhang, Jie Liu, Lei Qi, Zhiyuan Liu, and Maosong Sun. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems, 2024. URL https://arxiv.org/abs/2402.14008.

Jian Hu. Reinforce++: A simple and efficient approach for aligning large language models, 2025. URL https://arxiv.org/abs/2501.03262.

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-reasoner-zero: An open source approach to scaling up reinforcement learning on the base model, 2025. URL https://arxiv.org/abs/2503.24290.

Allan Jabri, Kyle Hsu, Ben Eysenbach, Abhishek Gupta, Sergey Levine, and Chelsea Finn. Unsupervised curricula for visual meta-reinforcement learning, 2019. URL https://arxiv.org/abs/1912.04226.

Niels Justesen, Ruben Rodriguez Torrado, Philip Bontrager, Ahmed Khalifa, Julian Togelius, and Sebastian Risi. Illuminating generalization in deep reinforcement learning through procedural level generation, 2018. URL https://arxiv.org/abs/1806.10729.

Amirhossein Kazemnejad, Milad Aghajohari, Eva Portelance, Alessandro Sordoni, Siva Reddy, Aaron Courville, and Nicolas Le Roux. Vineppo: Unlocking rl potential for llm reasoning through refined credit assignment, 2024. URL https://arxiv.org/abs/2410.01679.

Tomasz Korbak, Hady Elsahar, Germán Kruszewski, and Marc Dymetman. On reinforcement learning and distribution matching for fine-tuning language models with no catastrophic forgetting. *Advances in Neural Information Processing Systems*, 35:16203–16220, 2022.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models, 2022. URL https://arxiv.org/abs/2206.14858.

Xuefeng Li, Haoyang Zou, and Pengfei Liu. Limr: Less is more for rl scaling, 2025. URL https://arxiv.org/abs/2502.11886.

Ziniu Li, Tian Xu, Yushun Zhang, Zhihang Lin, Yang Yu, Ruoyu Sun, and Zhi-Quan Luo. Remax: A simple, effective, and efficient reinforcement learning method for aligning large language models, 2024. URL https://arxiv.org/abs/2310.10505.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step, 2023. URL https://arxiv.org/abs/2305.20050.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Li Erran Li, Raluca Ada Popa, and Ion Stoica. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl, 2025. Notion Blog.

Tambet Matiisen, Avital Oliver, Taco Cohen, and John Schulman. Teacher-student curriculum learning, 2017. URL https://arxiv.org/abs/1707.00183.

Niklas Muennighoff, Zitong Yang, Weijia Shi, Xiang Lisa Li, Li Fei-Fei, Hannaneh Hajishirzi, Luke Zettlemoyer, Percy Liang, Emmanuel Candès, and Tatsunori Hashimoto. s1: Simple test-time scaling, 2025. URL https://arxiv.org/abs/2501.19393.

OpenAI, :, Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Mądry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, Alex Paino, Alex Renzin, Alex Tachard Passos, Alexander Kirillov, Alexi Christakis, Alexis Conneau, Ali Kamali, Allan Jabri, Allison Moyer, Allison Tam, Amadou Crookes, Amin Tootoochian, Amin Tootoonchian, Ananya Kumar, Andrea Vallone, Andrej Karpathy, Andrew Braunstein, Andrew Cann, Andrew Codispoti, Andrew Galu, Andrew Kondrich, Andrew Tulloch, Andrey Mishchenko, Angela Baek, Angela Jiang, Antoine Pelisse, Antonia Woodford, Anuj Gosalia, Arka Dhar, Ashley Pantuliano, Avi Nayak, Avital Oliver, Barret Zoph, Behrooz Ghorbani, Ben Leimberger, Ben Rossen, Ben Sokolowsky, Ben Wang, Benjamin Zweig, Beth Hoover, Blake Samic, Bob McGrew, Bobby Spero, Bogo Giertler, Bowen Cheng, Brad Lightcap, Brandon Walkin, Brendan Quinn, Brian Guarraci, Brian Hsu, Bright Kellogg, Brydon Eastman, Camillo Lugaresi, Carroll Wainwright, Cary Bassin, Cary Hudson, Casey Chu, Chad Nelson, Chak Li, Chan Jun Shern, Channing Conger, Charlotte Barette, Chelsea Voss, Chen Ding, Cheng Lu, Chong Zhang, Chris Beaumont, Chris Hallacy, Chris Koch, Christian Gibson, Christina Kim, Christine Choi, Christine McLeavey, Christopher Hesse, Claudia Fischer, Clemens Winter, Coley Czarnecki, Colin Jarvis, Colin Wei, Constantin Koumouzelis, Dane Sherburn, Daniel Kappler, Daniel Levin, Daniel Levy, David Carr, David Farhi, David Mely, David Robinson, David Sasaki, Denny Jin, Dev Valladares, Dimitris Tsipras, Doug Li, Duc Phong Nguyen, Duncan Findlay, Edede Oiwoh, Edmund Wong, Ehsan Asdar, Elizabeth Proehl, Elizabeth Yang, Eric Antonow, Eric Kramer, Eric Peterson, Eric Sigler, Eric Wallace, Eugene Brevdo, Evan Mays, Farzad Khorasani, Felipe Petroski Such, Filippo Raso, Francis Zhang, Fred von Lohmann, Freddie Sulit, Gabriel Goh, Gene Oden, Geoff Salmon, Giulio Starace, Greg Brockman, Hadi Salman, Haiming Bao, Haitang Hu, Hannah Wong, Haoyu Wang, Heather Schmidt, Heather Whitney, Heewoo Jun, Hendrik Kirchner, Henrique Ponde de Oliveira Pinto, Hongyu Ren, Huiwen Chang, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian O'Connell, Ian Osband, Ian Silber, Ian Sohl, Ibrahim Okuyucu, Ikai Lan, Ilya Kostrikov, Ilya Sutskever, Ingmar Kanitscheider, Ishaan Gulrajani, Jacob Coxon, Jacob Menick, Jakub Pachocki, James Aung, James Betker, James Crooks, James Lennon, Jamie Kiros, Jan Leike, Jane Park, Jason Kwon, Jason Phang, Jason Teplitz, Jason Wei, Jason Wolfe, Jay Chen, Jeff Harris, Jenia Varavva, Jessica Gan Lee, Jessica Shieh, Ji Lin, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joanne Jang, Joaquin Quinonero Candela, Joe Beutler, Joe Landers, Joel Parish, Johannes Heidecke, John Schulman, Jonathan Lachman, Jonathan McKay, Jonathan Uesato, Jonathan Ward, Jong Wook Kim, Joost Huizinga, Jordan Sitkin, Jos Kraaijeveld, Josh Gross, Josh Kaplan, Josh Snyder, Joshua Achiam, Joy Jiao, Joyce Lee, Juntang Zhuang, Justyn Harriman, Kai Fricke, Kai Hayashi, Karan Singhal, Katy Shi, Kavin Karthik, Kayla Wood, Kendra Rimbach, Kenny Hsu, Kenny Nguyen, Keren Gu-Lemberg, Kevin Button, Kevin Liu, Kiel Howe, Krithika Muthukumar, Kyle Luther, Lama Ahmad, Larry Kai, Lauren Itow, Lauren Workman, Leher Pathak, Leo Chen, Li Jing, Lia Guy, Liam Fedus, Liang Zhou, Lien Mamitsuka, Lilian Weng, Lindsay McCallum, Lindsey Held, Long Ouyang, Louis Feuvrier, Lu Zhang, Lukas Kondraciuk, Lukasz Kaiser, Luke Hewitt, Luke Metz, Lyric Doshi, Mada Aflak, Maddie Simens, Madelaine Boyd, Madeleine Thompson, Marat Dukhan, Mark Chen, Mark Gray, Mark Hudnall, Marvin Zhang, Marwan Aljubeh, Mateusz Litwin, Matthew Zeng, Max Johnson, Maya Shetty, Mayank Gupta, Meghan Shah, Mehmet Yatbaz, Meng Jia Yang, Mengchao Zhong, Mia Glaese, Mianna Chen, Michael Janner, Michael Lampe, Michael Petrov, Michael Wu, Michele Wang, Michelle Fradin, Michelle Pokrass, Miguel Castro, Miguel Oom Temudo de Castro, Mikhail Pavlov, Miles Brundage, Miles Wang, Minal Khan, Mira Murati, Mo Bavarian, Molly Lin, Murat Yesildal, Nacho Soto, Natalia Gimelshein, Natalie Cone, Natalie Staudacher, Natalie Summers, Natan LaFontaine, Neil Chowdhury, Nick Ryder, Nick Stathas, Nick Turley, Nik Tezak, Niko Felix, Nithanth Kudige, Nitish Keskar, Noah Deutsch, Noel Bundick, Nora Puckett, Ofir Nachum, Ola Okelola, Oleg Boiko, Oleg Murk, Oliver Jaffe, Olivia Watkins, Olivier Godement, Owen Campbell-Moore, Patrick Chao, Paul McMillan, Pavel Belov, Peng Su, Peter Bak, Peter Bakkum, Peter Deng, Peter Dolan, Peter Hoeschele, Peter Welinder, Phil Tillet, Philip Pronin, Philippe Tillet, Prafulla Dhariwal, Qiming Yuan, Rachel Dias, Rachel Lim, Rahul Arora, Rajan Troll, Randall Lin, Rapha Gontijo Lopes, Raul Puri, Reah Miyara, Reimar Leike, Renaud Gaubert, Reza Zamani, Ricky Wang, Rob Donnelly, Rob Honsby, Rocky Smith, Rohan Sahai, Rohit Ramchandani, Romain Huet, Rory Carmichael, Rowan Zellers, Roy Chen, Ruby Chen, Ruslan Nigmatullin, Ryan Cheu, Saachi Jain, Sam Altman, Sam Schoenholz, Sam Toizer, Samuel Miserendino, Sandhini Agarwal, Sara Culver, Scott Ethersmith, Scott Gray, Sean Grove, Sean Metzger, Shamez Hermani, Shantanu Jain, Shengjia Zhao, Sherwin Wu, Shino Jomoto, Shirong Wu, Shuaiqi, Xia, Sonia Phene, Spencer

Papay, Srinivas Narayanan, Steve Coffey, Steve Lee, Stewart Hall, Suchir Balaji, Tal Broda, Tal Stramer, Tao Xu, Tarun Gogineni, Taya Christianson, Ted Sanders, Tejal Patwardhan, Thomas Cunninghman, Thomas Degry, Thomas Dimson, Thomas Raoux, Thomas Shadwell, Tianhao Zheng, Todd Underwood, Todor Markov, Toki Sherbakov, Tom Rubin, Tom Stasi, Tomer Kaftan, Tristan Heywood, Troy Peterson, Tyce Walters, Tyna Eloundou, Valerie Qi, Veit Moeller, Vinnie Monaco, Vishal Kuo, Vlad Fomenko, Wayne Chang, Weiyi Zheng, Wenda Zhou, Wesam Manassra, Will Sheu, Wojciech Zaremba, Yash Patil, Yilei Qian, Yongjik Kim, Youlong Cheng, Yu Zhang, Yuchen He, Yuchen Zhang, Yujia Jin, Yunxing Dai, and Yury Malkov. Gpt-4o system card, 2024a. URL https://arxiv.org/abs/2410.21276.

OpenAI, :, Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, Alex Iftimie, Alex Karpenko, Alex Tachard Passos, Alexander Neitz, Alexander Prokofiev, Alexander Wei, Allison Tam, Ally Bennett, Ananya Kumar, Andre Saraiva, Andrea Vallone, Andrew Duberstein, Andrew Kondrich, Andrey Mishchenko, Andy Applebaum, Angela Jiang, Ashvin Nair, Barret Zoph, Behrooz Ghorbani, Ben Rossen, Benjamin Sokolowsky, Boaz Barak, Bob McGrew, Borys Minaiev, Botao Hao, Bowen Baker, Brandon Houghton, Brandon McKinzie, Brydon Eastman, Camillo Lugaresi, Cary Bassin, Cary Hudson, Chak Ming Li, Charles de Bourcy, Chelsea Voss, Chen Shen, Chong Zhang, Chris Koch, Chris Orsinger, Christopher Hesse, Claudia Fischer, Clive Chan, Dan Roberts, Daniel Kappler, Daniel Levy, Daniel Selsam, David Dohan, David Farhi, David Mely, David Robinson, Dimitris Tsipras, Doug Li, Dragos Oprica, Eben Freeman, Eddie Zhang, Edmund Wong, Elizabeth Proehl, Enoch Cheung, Eric Mitchell, Eric Wallace, Erik Ritter, Evan Mays, Fan Wang, Felipe Petroski Such, Filippo Raso, Florencia Leoni, Foivos Tsimpourlas, Francis Song, Fred von Lohmann, Freddie Sulit, Geoff Salmon, Giambattista Parascandolo, Gildas Chabot, Grace Zhao, Greg Brockman, Guillaume Leclerc, Hadi Salman, Haiming Bao, Hao Sheng, Hart Andrin, Hessam Bagherinezhad, Hongyu Ren, Hunter Lightman, Hyung Won Chung, Ian Kivlichan, Ian O'Connell, Ian Osband, Ignasi Clavera Gilaberte, Ilge Akkaya, Ilya Kostrikov, Ilya Sutskever, Irina Kofman, Jakub Pachocki, James Lennon, Jason Wei, Jean Harb, Jerry Twore, Jiacheng Feng, Jiahui Yu, Jiayi Weng, Jie Tang, Jieqi Yu, Joaquin Quiñonero Candela, Joe Palermo, Joel Parish, Johannes Heidecke, John Hallman, John Rizzo, Jonathan Gordon, Jonathan Uesato, Jonathan Ward, Joost Huizinga, Julie Wang, Kai Chen, Kai Xiao, Karan Singhal, Karina Nguyen, Karl Cobbe, Katy Shi, Kayla Wood, Kendra Rimbach, Keren Gu-Lemberg, Kevin Liu, Kevin Lu, Kevin Stone, Kevin Yu, Lama Ahmad, Lauren Yang, Leo Liu, Leon Maksin, Leyton Ho, Liam Fedus, Lilian Weng, Linden Li, Lindsay McCallum, Lindsey Held, Lorenz Kuhn, Lukas Kondraciuk, Lukasz Kaiser, Luke Metz, Madelaine Boyd, Maja Trebacz, Manas Joglekar, Mark Chen, Marko Tintor, Mason Meyer, Matt Jones, Matt Kaufer, Max Schwarzer, Meghan Shah, Mehmet Yatbaz, Melody Y. Guan, Mengyuan Xu, Mengyuan Yan, Mia Glaese, Mianna Chen, Michael Lampe, Michael Malek, Michele Wang, Michelle Fradin, Mike McClay, Mikhail Pavlov, Miles Wang, Mingxuan Wang, Mira Murati, Mo Bavarian, Mostafa Rohaninejad, Nat McAleese, Neil Chowdhury, Neil Chowdhury, Nick Ryder, Nikolas Tezak, Noam Brown, Ofir Nachum, Oleg Boiko, Oleg Murk, Olivia Watkins, Patrick Chao, Paul Ashbourne, Pavel Izmailov, Peter Zhokhov, Rachel Dias, Rahul Arora, Randall Lin, Rapha Gontijo Lopes, Raz Gaon, Reah Miyara, Reimar Leike, Renny Hwang, Rhythm Garg, Robin Brown, Roshan James, Rui Shu, Ryan Cheu, Ryan Greene, Saachi Jain, Sam Altman, Sam Toizer, Sam Toyer, Samuel Miserendino, Sandhini Agarwal, Santiago Hernandez, Sasha Baker, Scott McKinney, Scottie Yan, Shengjia Zhao, Shengli Hu, Shibani Santurkar, Shraman Ray Chaudhuri, Shuyuan Zhang, Siyuan Fu, Spencer Papay, Steph Lin, Suchir Balaji, Suvansh Sanjeev, Szymon Sidor, Tal Broda, Aidan Clark, Tao Wang, Taylor Gordon, Ted Sanders, Tejal Patwardhan, Thibault Sottiaux, Thomas Degry, Thomas Dimson, Tianhao Zheng, Timur Garipov, Tom Stasi, Trapit Bansal, Trevor Creech, Troy Peterson, Tyna Eloundou, Valerie Qi, Vineet Kosaraju, Vinnie Monaco, Vitchyr Pong, Vlad Fomenko, Weiyi Zheng, Wenda Zhou, Wes McCabe, Wojciech Zaremba, Yann Dubois, Yinghai Lu, Yining Chen, Young Cha, Yu Bai, Yuchen He, Yuchen Zhang, Yunyun Wang, Zheng Shao, and Zhuohan Li. Openai o1 system card, 2024b. URL https://arxiv.org/abs/2412.16720.

Rémy Portelas, Cédric Colas, Katja Hofmann, and Pierre-Yves Oudeyer. Teacher algorithms for curriculum learning of deep rl in continuously parameterized environments, 2019. URL https://arxiv.org/abs/1910.07224.

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, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36:53728–53741, 2023.

Rafael Rafailov, Joey Hejna, Ryan Park, and Chelsea Finn. From r to q*: Your language model is secretly a q-function. *arXiv preprint arXiv:2404.12358*, 2024.

Pierre Harvey Richemond, Yunhao Tang, Daniel Guo, Daniele Calandriello, Mohammad Gheshlaghi Azar, Rafael Rafailov, Bernardo Avila Pires, Eugene Tarassov, Lucas Spangher, Will Ellsworth, et al. Offline regularised reinforcement learning for large language models alignment. *arXiv preprint arXiv:2405.19107*, 2024.

Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, and Raia Hadsell. Progressive neural networks, 2022. URL https://arxiv.org/abs/1606.04671.

John Schulman, Xi Chen, and Pieter Abbeel. Equivalence between policy gradients and soft q-learning. *arXiv preprint arXiv:1704.06440*, 2017a.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017b. URL https://arxiv.org/abs/1707.06347.

Guangming Sheng, Chi Zhang, Zilingfeng Ye, Xibin Wu, Wang Zhang, Ru Zhang, Yanghua Peng, Haibin Lin, and Chuan Wu. Hybridflow: A flexible and efficient rlhf framework. *arXiv preprint arXiv: 2409.19256*, 2024.

Mingyang Song, Mao Zheng, Zheng Li, Wenjie Yang, Xuan Luo, Yue Pan, and Feng Zhang. Fastcurl: Curriculum reinforcement learning with progressive context extension for efficient training r1-like reasoning models, 2025. URL https://arxiv.org/abs/2503.17287.

Sainbayar Sukhbaatar, Zeming Lin, Ilya Kostrikov, Gabriel Synnaeve, Arthur Szlam, and Rob Fergus. Intrinsic motivation and automatic curricula via asymmetric self-play, 2018. URL https://arxiv.org/abs/1703.05407.

Rui Wang, Joel Lehman, Jeff Clune, and Kenneth O. Stanley. Paired open-ended trailblazer (poet): Endlessly generating increasingly complex and diverse learning environments and their solutions, 2019. URL https://arxiv.org/abs/1901.01753.

Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Lucas Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, Weizhu Chen, Shuohang Wang, Simon Shaolei Du, and Yelong Shen. Reinforcement learning for reasoning in large language models with one training example, 2025. URL https://arxiv.org/abs/2504.20571.

Liang Wen, Yunke Cai, Fenrui Xiao, Xin He, Qi An, Zhenyu Duan, Yimin Du, Junchen Liu, Lifu Tang, Xiaowei Lv, Haosheng Zou, Yongchao Deng, Shousheng Jia, and Xiangzheng Zhang. Light-r1: Curriculum sft, dpo and rl for long cot from scratch and beyond, 2025. URL https://arxiv.org/abs/2503.10460.

Yixin Ye, Zhen Huang, Yang Xiao, Ethan Chern, Shijie Xia, and Pengfei Liu. Limo: Less is more for reasoning, 2025. URL https://arxiv.org/abs/2502.03387.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Weinan Dai, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Mu Qiao, Yonghui Wu, and Mingxuan Wang. Dapo: An open-source llm reinforcement learning system at scale, 2025. URL https://arxiv.org/abs/2503.14476.

Wojciech Zaremba and Ilya Sutskever. Learning to execute, 2015. URL https://arxiv.org/abs/1410.4615.

Andrew Zhao, Yiran Wu, Yang Yue, Tong Wu, Quentin Xu, Yang Yue, Matthieu Lin, Shenzhi Wang, Qingyun Wu, Zilong Zheng, and Gao Huang. Absolute zero: Reinforced self-play reasoning with zero data, 2025. URL https://arxiv.org/abs/2505.03335.

Zyphra. Zr1-1.5b, a small but powerful reasoning model for math and code, 2025. URL https://www.zyphra.com/post/introducing-zr1-1-5b-a-small-but-powerful-math-code-reasoning-model.

A Related Work

Efficient Reinforcement Finetuning. Most RFT pipelines build on Proximal Policy Optimization (PPO) (Schulman et al., 2017b), with recent variants like RAFT (Dong et al., 2023), ReMax (Li et al., 2024), GRPO (DeepSeek-AI et al., 2025), and REINFORCE++ (Hu, 2025), aiming to reduce computational overhead by simplifying RL components. While effective, these methods often trade off stability or sample efficiency. In parallel, data-centric strategies have emerged as promising alternatives for efficient finetuning. LIMO (Ye et al., 2025) and s1 (Muennighoff et al., 2025) show that small, carefully selected supervised datasets can yield strong downstream performance, but their success hinges on manual curation, prompt engineering, and careful dataset construction, which may not generalize across tasks or models. LIMR (Li et al., 2025) and Wang et al. (2025) proposes scoring training examples based on their estimated learning impact, enabling selective finetuning with fewer samples. Yet, computing these scores requires a full training run, and the scores must be recomputed for each new model, limiting practicality and scalability. Moreover, reducing the number of training samples does not inherently translate to improved efficiency. Models still require a comparable number of optimization steps and wall-clock time to converge. In contrast, ADARFT introduces a lightweight, model-agnostic curriculum learning strategy that dynamically adjusts task difficulty based on reward feedback. This allows continuous adaptation to the model's capabilities, improving convergence speed and final accuracy without modifying the RL algorithm or requiring manual data curation.

Curriculum Learning for RL. Curriculum learning (CL) structures training by presenting tasks in an organized progression, typically from easy to hard, to enhance learning efficiency and generalization (Bengio et al., 2009). In RL, CL methods include task sorting by difficulty (Zaremba & Sutskever, 2015; Justesen et al., 2018; Wang et al., 2019), teacher-student frameworks that adaptively select tasks based on learning progress (Matiisen et al., 2017; Portelas et al., 2019), and self-play approaches that induce automatic curricula through agent competition (Sukhbaatar et al., 2018; Zhao et al., 2025). Other strategies use intermediate-goal generation in sparse-reward settings (Florensa et al., 2018), unsupervised skill discovery (Jabri et al., 2019), or knowledge transfer via progressive networks and imitation (Czarnecki et al., 2018; Rusu et al., 2022). While CL is well-studied in classical RL, its application to RFT of LLMs is still limited. Existing methods typically use staged training with hand-designed difficulty tiers (Wen et al., 2025; Luo et al., 2025; Song et al., 2025), or online filtering schemes that repeatedly sample and discard data until rewards reach a target range (Bae et al., 2025; Yu et al., 2025). These methods either lack adaptability or introduce significant computational overhead due to repeated rollouts. In contrast, ADARFT is among the first truly adaptive curriculum learning approaches for RFT: it continuously adjusts task difficulty based on the model's reward signal, enabling efficient, scalable training without fixed schedules or repeated rollouts.

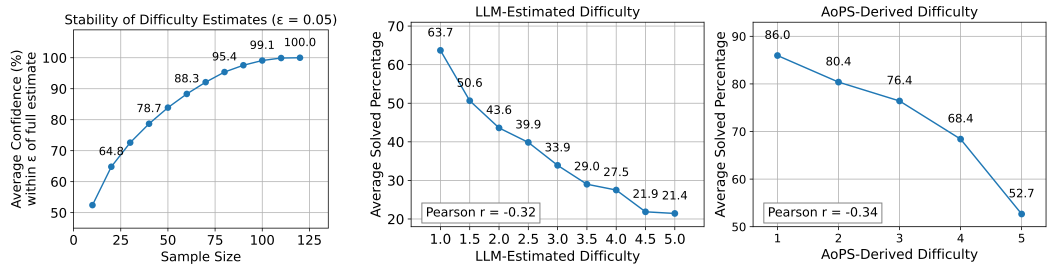
B Implementation Details of AdaRFT

B.1 Dynamic Curriculum Sampling

To construct an adaptive curriculum, we define a target difficulty T, which represents the current target difficulty level for training (more in § B.3). ADARFT dynamically adjusts T based on the model's reward signal to maintain an optimal difficulty level for learning. At each step, the algorithm computes the absolute difference between the target difficulty and the difficulty of each problem in the dataset (Alg. 1, line 4): $\Delta_i = |d_i - T|$ for all $i \in [1, |D|]$. The batch of training problems is formed by selecting the B problems with the smallest values of Δ_i (Alg. 1, line 5), producing a batch: $X = \{s_1, s_2, \ldots, s_B\}$. This ensures that the selected problems are closest to the model's current target difficulty, focusing the learning process on problems that are neither too easy nor too hard.

B.2 Policy Update

The selected batch X is used to train the policy model π_θ, which generates responses: $G = \pi_\theta(X)$. A reward signal is computed based on the correctness of the model's output (Alg. 1, line 7): $R_i = 1$ if the response is correct, and $R_i = 0$ if the response is incorrect. The average reward over the batch is computed as (Alg. 1, line 7): $R_{avg} = \frac{1}{|X|} \sum_{i=1}^{|X|} R(X_i, G_i)$. The policy can then be updated



(a) Average confidence that subsampled difficulty estimates fall within ±0.05 of the full-sample estimate.

(b) Correlation between average solved percentage and two types of difficulty labels: (left) LLM-estimated difficulty and (right) AoPS-derived difficulty levels.

Figure 2: Evaluation of difficulty estimation: (a) Stability of difficulty scores under subsampling of model rollouts; (b) Correlation between labeled difficulty levels and average solved percentage.

using a reinforcement learning algorithm $\mathcal{A}$ such as PPO, GRPO, or REINFORCE++ (Alg. 1, line 8): $\pi_\theta \leftarrow \mathcal{A}(\pi_\theta, X, G, R)$.

B.3 Target Difficulty Update

To adapt the curriculum dynamically, the target difficulty is updated based on the average reward. If the model performs well on the current difficulty level (high reward), the target difficulty increases, making the training problems harder. Conversely, if the model performs poorly, the target difficulty decreases. This dynamic update mechanism lies at the core of ADARFT's curriculum adaptation strategy. The update rule (Alg. 1, line 9) is defined as:

$$T' = \text{clip}(T + \eta \cdot \tanh(\alpha \cdot (R_{avg} - \beta)), d_{\min}, d_{\max})$$

Here, η, α, β are hyperparameters: η is the step size for adjusting the target difficulty, α controls the sensitivity of the update, and β is the target reward level, representing the desired success rate. The $\tanh$ function ensures smooth updates and prevents large jumps in difficulty by saturating for large deviations, while the "clip" function constrains the target difficulty within the valid range $[d_{\min}, d_{\max}]$. These bounds can be manually specified or automatically derived from the training set, for example by taking the minimum and maximum of the difficulty scores $\{d_i\}$. Intuition and guidance for selecting these hyperparameters are discussed in Section D.1 and 3.3.

C Stability and Reliability of Difficulty Estimation

To evaluate the stability of our difficulty estimation process, we simulate how confidence varies with different numbers of samples. For each problem, we treat the full set of 128 rollouts as the ground-truth difficulty estimate and compute how often sub-sampled estimates fall within a tolerance of $\epsilon = 0.05$. Specifically, we run 10 random sampling trials per sample size and average the confidence across all problems in the dataset. As shown in Figure 2a, even with as few as 64 samples, the estimated difficulty remains within ± 0.05 of the full estimate over 90% of the time. With just 40 samples, the confidence remains around 80%. These results indicate that accurate and robust difficulty estimation can be achieved with significantly fewer rollouts, reducing the computational burden of large-scale curriculum construction.

To further validate the reliability of our difficulty estimates, we examined their alignment with the difficulty levels provided in the MATH dataset. The MATH dataset comprises 12,500 competition-level mathematics problems sourced from contests such as the American Mathematics Competitions (AMC) and the American Invitational Mathematics Examination (AIME). Each problem is categorized into one of five difficulty levels, following the classification system used by the Art of Problem Solving (AoPS) community.[1] In this system, level 1 denotes the easiest problems, while level 5 represents the most difficult. As shown in Figure 2b, there is a clear downward trend in the average solve rate as the labeled difficulty level increases, ranging from 86.0% at level 1 to 52.7% at level 5. Specifically,

[1] `https://artofproblemsolving.com/wiki/index.php/AoPS_Wiki:Competition_ratings`

the AoPS-derived difficulty levels yield a Pearson correlation of $r = -0.34$ ($p < 0.05$) with model success rates. This negative correlation indicates that the model's empirical performance aligns well with the intended difficulty stratification, reinforcing the utility of both the labeled difficulty levels and our estimation approach in guiding curriculum learning. To further streamline the difficulty estimation process, we also prompted GPT-4o (`gpt-4o-0806`) (OpenAI et al., 2024a) to assign difficulty levels to the DeepScaleR dataset based on the AoPS rubric. Each problem was presented to GPT-4o with a request to rate its difficulty according to AoPS guidelines (the full prompt is shown in Appendix E.3). This approach provides a lightweight and scalable alternative to rollout-based estimation. As shown in Figure 2b, GPT-4o's difficulty ratings also correlate well with the model success rates, with a Pearson correlation of $r = -0.32$ ($p < 0.05$), making it a practical proxy for curriculum scheduling when computational resources are constrained.

D Further Discussion

D.1 Theoretical Justification for Target Reward β

A key component of AdaRFT is its adaptive curriculum mechanism, which steers training toward a target reward level β. Intuitively, we aim to train on examples that are neither trivially easy nor prohibitively hard. In this light, setting $\beta = 0.5$, corresponding to a success rate of roughly 50%, naturally aligns with this goal. This section formalizes that intuition by analyzing the relationship between reward variance and learnability in RFT with binary rewards.

In entropy-regularized reinforcement learning, the optimal policy π^* can be expressed relative to a reference policy π_{init} as (Korbak et al., 2022; Go et al., 2023; Rafailov et al., 2023):

$$\pi^*(y \mid x) = Z(x)\pi_{\text{init}}(y \mid x) \exp\left(\frac{1}{\tau} r(x, y)\right) \tag{1}$$

where τ is the inverse temperature parameter controlling entropy regularization, and $Z(x)$ is the partition function that normalizes the action probability. The corresponding optimal value function and the partition function is given by (Schulman et al., 2017a; Richemond et al., 2024):

$$V^*(x) := \tau \log \mathbb{E}_{y \sim \pi_{\text{init}}(\cdot|x)} \left[\exp\left(\frac{1}{\tau} r(x, y)\right)\right] \quad \text{and} \quad Z(x) = \exp\left(\frac{1}{\tau} V^*(x)\right) \tag{2}$$

We can then take the expectation of the log-ratio between the optimal policy and the initial policy with respect to $y \sim \pi_{\text{init}}(\cdot \mid x)$, leading to (Haarnoja et al., 2017; Schulman et al., 2017a):

$$\mathbb{E}_{y \sim \pi_{\text{init}}(\cdot|x)} \left[\log \frac{\pi^*(y \mid x)}{\pi_{\text{init}}(y \mid x)}\right] = \frac{1}{\tau} \mathbb{E}_{\pi_{\text{init}}}[r(x, y)] - \frac{1}{\tau} V^*(x) \tag{3}$$

Since the left-hand side can be interpreted as the negative reverse KL divergence between π_{init} and π^* (Rafailov et al., 2024), Bae et al. (2025) show that when the reward $r(x, y)$ with $y \sim \pi_{\text{init}}(\cdot \mid x)$ is Bernoulli, the KL divergence is lower-bounded by the reward variance:

$$D_{\text{KL}}(\pi_{\text{init}} \| \pi^*) \geq \frac{p(x)(1 - p(x))}{2\tau^2} \tag{4}$$

where $p(x)$ is the model's success rate on prompt x. This implies that the lower bound on the KL divergence, and consequently the gradient magnitude during policy updates, is proportional to the reward variance, which is maximized when $p(x) = 0.5$. In other words, training on prompts that the model succeeds on roughly half the time may yield the strongest learning signal. In Section 4 and Appendix D.3, we conduct an ablation study by varying the target reward β, demonstrating that setting $\beta = 0.5$ consistently leads to the best performance, supporting the hypothesis that training on prompts with a success rate near 50% provides the most informative learning signal.

D.2 Data Difficulty on Model Performance

To better understand the effect of data difficulty on model performance, we introduce two additional data distributions: easy-extreme and hard-extreme. Unlike the skew-difficult and skew-easy distributions, which still include a mix of difficulty levels, the easy-extreme and hard-extreme sets consist exclusively of the most polarized examples. Specifically, easy-extreme contains only the easiest samples with difficulty levels no greater than 15, while hard-extreme includes only the hardest samples with difficulty levels of at least 97. Each of these extreme distributions consists of approximately 8,000 samples, providing a focused and controlled evaluation of model behavior under minimal or maximal difficulty conditions. We trained a Qwen 2.5 7B model on each of the two extreme distributions using PPO, and compared their performance to models trained on the uniform distribution with PPO (Uniform) and with ADARFT instantiated with PPO (Uniform + ADARFT), as described in Section 4. The results are presented in Figure 3. The key takeaway is that training on only overly easy or hard problems fails to provide useful learning signals, reinforcing the need for ADARFT to adaptively steer models toward challenges matched to their current ability.

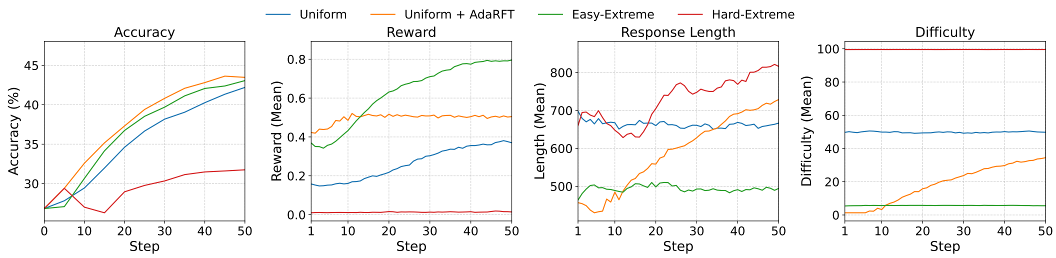


Figure 3: Performance comparison of Qwen 2.5 7B trained on different data distributions using PPO (Uniform, Easy-Extreme, Hard-Extreme) and ADARFT instantiated with PPO (Uniform + ADARFT). For clarity, curves are exponentially smoothed ($\alpha = 0.3$) to reduce noise.

Accuracy. The leftmost panel of Figure 3 shows that uniform + ADARFT achieves the highest overall accuracy throughout training, outperforming both uniform and the two extreme settings. This highlights the effectiveness of ADARFT in guiding the model through an optimal difficulty progression. In contrast, hard-extreme struggles significantly, with a flat and lower trajectory, indicating that exposing the model only to very difficult problems limits learning progress. This suggests that without a gradual exposure strategy, models trained on only the hardest problems are unable to bootstrap their capabilities effectively.

Reward. The reward trends provide important clues about learning dynamics. The easy-extreme setup achieves the fastest reward improvement during early training, surpassing both uniform and hard-extreme. In particular, easy-extreme consistently operates in a reward range between 0.4 and 0.6 during early training, which corresponds to a success rate that is both challenging and attainable. In contrast, the reward of the uniform and hard-extreme setup lingers below 0.2 in early training, leading to slower learning. This suggests that training on problems with intermediate difficulty—those that are neither trivially easy nor prohibitively hard—provides the most effective learning signal. Notably, ADARFT is explicitly designed to exploit this insight: by setting the target reward $\beta = 0.5$, we encourage the model to train on problems that match this "productive struggle" zone. As shown by the uniform + ADARFT curve, the algorithm successfully maintains an average reward near 0.5 throughout training, allowing the model to learn at an optimal pace. Notably, while the uniform setup eventually reaches a reward of nearly 0.5 by step 50, it does not result in faster learning. This is likely because the model is already fairly well trained by that stage, so the additional reward signal contributes less to further improvement. In contrast, the hard-extreme model receives almost no reward signal for most of the training, while the uniform setup shows slower and more gradual reward accumulation.

Response Length. The response length panel reveals how the complexity of generated solutions evolves during training. The hard-extreme model consistently produces the longest responses, with length increasing steadily, reflecting the higher complexity and reasoning depth required by the hardest problems. In contrast, the easy-extreme setup maintains short and stable responses, consistent with its simpler problem set. The uniform and uniform + ADARFT setups fall between these two extremes. Notably, uniform + ADARFT shows a gradual increase in response length over time. This trend aligns with the behavior of the curriculum learning algorithm: as the model improves, it is exposed to increasingly difficult problems, which naturally demand more elaborate reasoning and longer solutions. This dynamic suggests that response length can serve as a useful proxy for problem difficulty and reasoning complexity during training.

Difficulty. Finally, the difficulty panel illustrates how problem difficulty evolves under each setup. The easy-extreme and hard-extreme curves remain flat, confirming that these datasets contain only problems from the tail ends of the difficulty spectrum (i.e., ≤ 15 and ≥ 97, respectively). The uniform curve is centered around 50, as expected, while uniform + ADARFT shows a steady increase in difficulty over time. This adaptive progression confirms that curriculum learning effectively steers the model from easier to harder problems, aligning difficulty with the model's evolving capabilities.

D.3 Ablation on Target Reward β

To better understand the role of the target reward β in ADARFT, we perform an ablation study varying β in the target difficulty update rule. Recall that β controls the target average reward the model is expected to achieve and implicitly steers the curriculum: lower values prioritize easier problems, while higher values shift the curriculum toward more challenging samples. We train a Qwen 2.5 Math 1.5B model on the uniform data distribution with ADARFT (PPO) using three different values of β: 0.2, 0.5, and 0.8. For comparison, we also include standard PPO without ADARFT (denoted as "w/o ADARFT") as a baseline.

As shown in Figure 4, the model trained with $\beta = 0.5$ achieves the highest accuracy throughout training. This supports our theoretical motivation in Section D.1: maximizing reward variance, which occurs when success rate ≈ 0.5, provides the strongest learning signal. Models with $\beta = 0.2$ and $\beta = 0.8$ underperform likely due to curriculum misalignment: $\beta = 0.8$ overly focuses on easy problems, while $\beta = 0.2$ overemphasizes difficult ones, both of which limit the model's capacity to generalize. The reward and difficulty curves align with the accuracy outcomes discussed above. The $\beta = 0.5$ configuration maintains a stable reward near 0.5, reflecting balanced difficulty exposure. In contrast, $\beta = 0.8$ results in overly high reward (i.e., easy samples), while $\beta = 0.2$ maintains a reward around 0.2 for most of training, indicating the model is repeatedly presented with overly difficult problems. As expected, response length is the shortest for $\beta = 0.8$ and longest for $\beta = 0.2$, consistent with the idea that longer responses correlate with problem complexity.

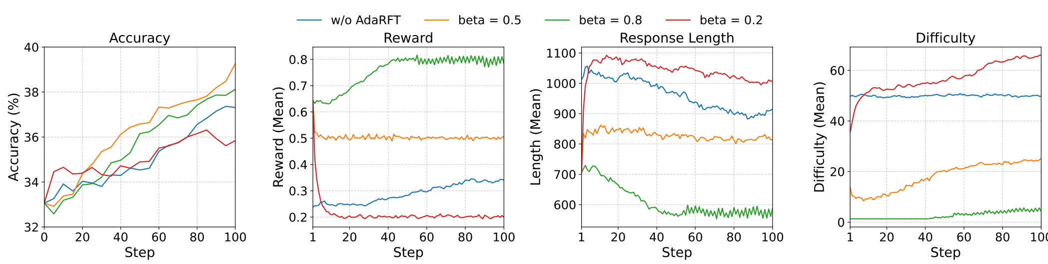


Figure 4: Ablation on β in ADARFT: we compare model accuracy, average reward, response length, and mean difficulty under $\beta = 0.2$, $\beta = 0.5$, and $\beta = 0.8$, along with standard PPO (w/o ADARFT).

D.4 Training on LLM-Estimated Difficulty

In addition to rollout-based difficulty estimation, we explore an alternative strategy that uses LLM-judged difficulty levels to guide curriculum construction. As described in Section 3.1, we prompt GPT-4o (`gpt-4o-0806`) to assign difficulty levels to math problems in the DeepScaleR dataset according to the AoPS rubric. This approach offers a lightweight and scalable alternative to computing pass@k success rates from model rollouts, making it especially attractive in low-resource scenarios.

To assess the effectiveness of this strategy, we train a Qwen 2.5 Math 1.5B model on the skew-difficult distribution using ADARFT (PPO) with two curriculum schedules: one based on rollout-derived pass@k difficulty, and the other guided by GPT-4o's difficulty ratings. Since the LLM-judged difficulty is on a scale of 1 to 5 (rather than 0 to 100), we set the step size hyperparameter $\eta = 2.5$ to align the difficulty adjustment magnitude with the reward signal. All other hyperparameters are kept unchanged. As shown in Figure 5, both curriculum strategies outperform

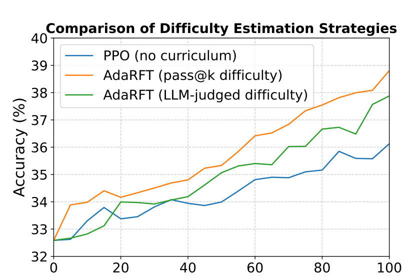


Figure 5: Comparison of different difficulty estimation strategies.

standard PPO without curriculum learning. While rollout-based difficulty estimation yields the strongest gains, the LLM-judged curriculum still provides a noticeable improvement over the baseline.

These results demonstrate that AdaRFT remains effective even when the difficulty signal is derived from heuristic or approximate sources like LLM judgments. Although less precise than empirical pass@k metrics, the LLM-based difficulty still provides enough structure to enable meaningful curriculum adaptation. This makes it a practical fallback when rollout computation is too costly, and suggests that future work could explore hybrid approaches that combine lightweight heuristics with periodic empirical calibration.

D.5 AdaRFT with Diverse RL Algorithms

To evaluate the generality of AdaRFT beyond PPO, we trained the Qwen 2.5 Math 1.5B model on a skew-difficult data distribution using two alternative reinforcement learning algorithms: REINFORCE++ and GRPO (see implementation details in Appendix E). As shown in Figure 6, AdaRFT significantly improves both the convergence speed and final accuracy across these variants. Across both cases, the adaptive curriculum acts orthogonally to the underlying optimization method. These results reinforce the plug-and-play nature of AdaRFT: it consistently enhances sample efficiency and policy robustness across algorithmic choices, making it broadly applicable in diverse reinforcement finetuning pipelines. Notably, this generalization holds without any additional tuning or algorithm-specific modifications, underscoring the practical utility of curriculum-aware training in both lightweight and computation-heavy RFT settings.

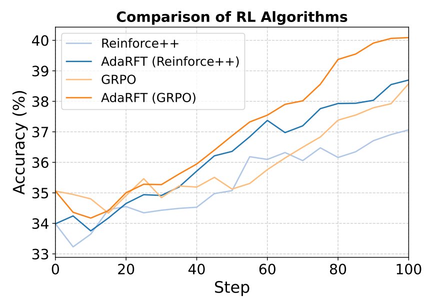


Figure 6: Comparison between models trained with and without AdaRFT using REINFORCE++ and GRPO.

E Implementation Details

E.1 Training Configuration

We trained both the actor and critic models using the PPO algorithm on a single node with 8 A100 GPUs. Each model was trained for approximately 100 optimization steps using the `veRL` library (Sheng et al., 2024). We used two model variants: `Qwen2.5-7B` and `Qwen2.5-MATH-1.5B`. The latter has a shorted context window, so we adjusted the max response length and the sequence parallel size accordingly.

Table 3 summarizes the core hyperparameter settings used across all three algorithms: PPO, GRPO, and REINFORCE++. We highlight both shared defaults and algorithm-specific overrides, including KL treatment modes, rollout settings, and critic configurations.

Following the approach of prior work (Bae et al., 2025; Hu et al., 2025; Zyphra, 2025), we perform a pass@40 analysis for each model and data distribution combination. For the data filtering baseline (3), we remove examples that are either too easy or too hard, excluding problems with solved percentages $\leq 10\%$ or $\geq 90\%$. This focuses training on problems of intermediate difficulty. However, this filtering discards a significant portion of the training data, including many potentially useful examples. Moreover, because difficulty is determined via pass@k metrics, filtering must be recomputed each time the model or data distribution changes. In contrast, AdaRFT only requires difficulty estimation once per dataset and adapts to any model during training. For AdaRFT, the target difficulty was dynamically adjusted throughout training based on the model's reward signal. This adjustment ensured that the model consistently encountered problems that matched its current skill level, preventing the model from being overwhelmed by overly difficult problems or stagnating on problems that were too easy.

E.2 AdaRFT Parameters

To enable effective AdaRFT, we incorporated a simple curriculum learning mechanism that dynamically adjusts task difficulty during training. We tuned several key hyperparameters: training batch size $B = 1024$, target reward $\beta = 0.5$, sensitivity parameter $\alpha = 2$, step size $\eta = 50$, and initial difficulty $T = 0$. These settings were used uniformly across all experiments listed in Table 3. The difficulty is updated as a function of the discrepancy between the running average reward $\bar{r}$ and the target reward β, using the following rule:

$$\Delta T = \eta \cdot \tanh\left(\alpha \cdot (\bar{r} - \beta)\right)$$

This formulation promotes stable learning by providing approximately linear updates when $\bar{r} \approx \beta$, and saturated updates when rewards deviate significantly, thereby avoiding abrupt difficulty shifts. Because the reward is bounded in $[0, 1]$ and the difficulty metric spans $[0, 100]$, we set the step size $\eta = 50$ to align their scales. The modulation parameter $\alpha = 2$ ensures smooth and controlled progression throughout training.

Table 3: Comparison of training hyperparameters for PPO, GRPO, and REINFORCE++ using the veRL library. Shared defaults are used unless overridden.

Category	**Parameter**	**PPO**	**GRPO**	**REINFORCE++**
Algorithm-Specific Settings				
General	Advantage estimator	GAE	GRPO	REINFORCE++
	Gamma (γ)	1.0	—	—
	Lambda (λ)	1.0	—	—
	Batch size	1024	1024	1024
	Max prompt length	1024	1024	1024
	Gradient checkpointing	Enabled	Enabled	Enabled
Actor	Learning rate	1×10^{-6}	1×10^{-6}	1×10^{-6}
	Mini-batch size	1024	1024	1024
	Dynamic batch size	Enabled	Enabled	Enabled
	KL penalty role	Reward	Loss	Loss
	KL loss type	Fixed	Low-variance KL	MSE
	KL loss coefficient (β)	0.001	0.001	0.001
	Entropy coefficient	0.001	0.001	0
	Clip ratio	0.2	0.2	0.2
	Gradient clipping	1.0	1.0	1.0
	Sequence parallel size	Model-specific	Model-specific	Model-specific
Rollout	Backend	vLLM	vLLM	vLLM
	Tensor model parallel size	2	2	2
	Rollouts per sample	1	4	1
	Nucleus sampling p	1.0	1.0	1.0
	GPU memory utilization	0.5	0.5	0.5
	Sampling temperature	1.0	1.0	1.0
Critic	Warmup steps	0	—	—
	Learning rate	1×10^{-5}	—	—
	Sequence parallel size	Model-specific	—	—
Model-Specific Overrides (shared across all algorithms)				
Qwen2.5-7B	Max response length	8000	8000	8000
	Sequence parallel size	2	2	2
	Max token length / GPU	8000	8000	8000
Qwen2.5-MATH-1.5B	Max response length	3000	3000	3000
	Sequence parallel size	1	1	1
	Max token length / GPU	16000	16000	16000
AdaRFT *Parameters*				
Curriculum Learning	Target reward (β)	0.5	0.5	0.5
	Sensitivity (α)	2	2	2
	Step size (η)	50	50	50
	Initial difficulty (T)	0	0	0

E.3 Prompt for Difficulty Estimation Using LLM as a Judge

The prompt used for difficulty estimation (as described in Section 3.1) is shown in Table 4, Table 5, and Table 6. The descriptions of the difficulty scales and examples are sourced from the AoPS Wiki.[2] Although GPT-4o is prompted to rate problem difficulty on a scale from 1 to 10, we found that over 95% of the problems fall within the range of 1 to 5. Therefore, we clip the scores and use a revised scale from 1 to 5. In addition to integer scores, we also allow half-point increments such as 1.5, 2.0, and 2.5 for finer-grained difficulty estimation.

[2] https://artofproblemsolving.com/wiki/index.php/AoPS_Wiki:Competition_ratings

Prompt for Difficulty Estimation (Part 1)

 # Math Problem
{problem}

 # Your Task
You are a subject matter expert in mathematics tasked with evaluating the difficulty level of individual math problems. Your assessment should be objective and based on a detailed difficulty scale provided below. Your judgment will help calibrate and categorize problems for use in educational settings or assessments. Be thorough, fair, and consistent in your evaluation.

 # Difficulty Scale
1: Problems strictly for beginner, on the easiest elementary school or middle school levels (MOEMS, MATHCOUNTS School, AMC 8 1-10, AMC 10 1-10, easier AMC 12 1-5, and others that involve standard techniques introduced up to the middle school level), most traditional middle/high school word problems.
1.5: Problems for stronger beginner students, on the level of the middling problems in most middle school contests (AMC 8 11-20, harder AMC 10 1-10, AMC 12 1-5, and those others that force students to apply their school-level knowledge to slightly more challenging problems), traditional middle/high school word problems with more complex problem solving.
2: For motivated beginners, harder questions from the previous categories (AMC 8 21-25, MATHCOUNTS Chapter (Sprint 21-30, Target 6-8), MATHCOUNTS States/Nationals, AMC 10 11-15, AMC 12 5-10, easiest AIME 1-3)
2.5: More advanced beginner problems, hardest questions from previous categories (Harder AMC 8 21-25, harder MATHCOUNTS States questions, AMC 10 16-20, AMC 12 11-15, usual AIME 1-3)
3: Early intermediate problems that require more creative thinking (harder MATHCOUNTS National questions, AMC 10 21-25, AMC 12 15-20, hardest AIME 1-3, usual AIME 4-6).
4: Intermediate-level problems (AMC 12 21-25, hardest AIME 4-6, usual AIME 7-10).
5: More difficult AIME problems (11-13), simple proof-based Olympiad-style problems (early JBMO questions, easiest USAJMO 1/4).
6: High-leveled AIME-styled questions (14/15). Introductory-leveled Olympiad-level questions (harder USAJMO 1/4 and easier USAJMO 2/5, easier USAMO and IMO 1/4).
7: Tougher Olympiad-level questions, may require more technical knowledge (harder USAJMO 2/5 and most USAJMO 3/6, extremely hard USAMO and IMO 1/4, easy-medium USAMO and IMO 2/5).
8: High-level Olympiad-level questions (medium-hard USAMO and IMO 2/5, easiest USAMO and IMO 3/6).
9: Expert Olympiad-level questions (average USAMO and IMO 3/6).
9.5: The hardest problems appearing on Olympiads which the strongest students could reasonably solve (hard USAMO and IMO 3/6).
10: Historically hard problems, generally unsuitable for very hard competitions (such as the IMO) due to being exceedingly tedious, long, and difficult (e.g. very few students are capable of solving on a worldwide basis).

Table 4: Prompt for difficulty estimation using LLM as a judge.

Prompt for Difficulty Estimation (Part 2)

 # Examples
For reference, here are some sample problems from each of the difficulty levels 1-10:
<1: Jamie counted the number of edges of a cube, Jimmy counted the numbers of corners, and Judy counted the number of faces. They then added the three numbers. What was the resulting sum? (2003 AMC 8, Problem 1)
1: How many integer values of x satisfy $|x| < 3\pi$? (2021 Spring AMC 10B, Problem 1)
1.5: A number is called flippy if its digits alternate between two distinct digits. For example, 2020 and 37373 are flippy, but 3883 and 123123 are not. How many five-digit flippy numbers are divisible by 15? (2020 AMC 8, Problem 19)
2: A fair 6-sided die is repeatedly rolled until an odd number appears. What is the probability that every even number appears at least once before the first occurrence of an odd number? (2021 Spring AMC 10B, Problem 18)
2.5: A, B, C are three piles of rocks. The mean weight of the rocks in A is 40 pounds, the mean weight of the rocks in B is 50 pounds, the mean weight of the rocks in the combined piles A and B is 43 pounds, and the mean weight of the rocks in the combined piles A and C is 44 pounds. What is the greatest possible integer value for the mean in pounds of the rocks in the combined piles B and C? (2013 AMC 12A, Problem 16)
3: Triangle ABC with $AB = 50$ and $AC = 10$ has area 120. Let D be the midpoint of $\overline{AB}$, and let E be the midpoint of $\overline{AC}$. The angle bisector of $\angle BAC$ intersects $\overline{DE}$ and $\overline{BC}$ at F and G, respectively. What is the area of quadrilateral $FDBG$? (2018 AMC 10A, Problem 24)
3.5: Find the number of integer values of k in the closed interval $[-500, 500]$ for which the equation $\log(kx) = 2\log(x + 2)$ has exactly one real solution. (2017 AIME II, Problem 7)
4: Define a sequence recursively by $x_0 = 5$ and

$$x_{n+1} = \frac{x_n^2 + 5x_n + 4}{x_n + 6}$$

for all nonnegative integers n. Let m be the least positive integer such that

$$x_m \leq 4 + \frac{1}{2^{20}}.$$

In which of the following intervals does m lie?
(A) $[9, 26]$ **(B)** $[27, 80]$ **(C)** $[81, 242]$ **(D)** $[243, 728]$ **(E)** $[729, \infty)$ (2019 AMC 10B, Problem 24 and 2019 AMC 12B, Problem 22)
4.5: Find, with proof, all positive integers n for which $2^n + 12^n + 2011^n$ is a perfect square. (USAJMO 2011/1)
5: Find all triples (a, b, c) of real numbers such that the following system holds:

$$a + b + c = \frac{1}{a} + \frac{1}{b} + \frac{1}{c},$$
$$a^2 + b^2 + c^2 = \frac{1}{a^2} + \frac{1}{b^2} + \frac{1}{c^2}.$$

(JBMO 2020/1)
5.5: Triangle ABC has $\angle BAC = 60°$, $\angle CBA \leq 90°$, $BC = 1$, and $AC \geq AB$. Let H, I, and O be the orthocenter, incenter, and circumcenter of $\triangle ABC$, respectively. Assume that the area of pentagon $BCOIH$ is the maximum possible. What is $\angle CBA$? (2011 AMC 12A, Problem 25)
6: Let $\triangle ABC$ be an acute triangle with circumcircle ω, and let H be the intersection of the altitudes of $\triangle ABC$. Suppose the tangent to the circumcircle of $\triangle HBC$ at H intersects ω at points X and Y with $HA = 3, HX = 2$, and $HY = 6$. The area of $\triangle ABC$ can be written in the form $m\sqrt{n}$, where m and n are positive integers, and n is not divisible by the square of any prime. Find $m + n$. (2020 AIME I, Problem 15)

Table 5: Prompt for difficulty estimation using LLM as a judge.

Prompt for Difficulty Estimation (Part 3)

6.5: Rectangles BCC_1B_2, CAA_1C_2, and ABB_1A_2 are erected outside an acute triangle ABC. Suppose that

$$\angle BC_1C + \angle CA_1A + \angle AB_1B = 180^\circ.$$

Prove that lines B_1C_2, C_1A_2, and A_1B_2 are concurrent. (USAMO 2021/1, USAJMO 2021/2)
7: We say that a finite set $\mathcal{S}$ in the plane is balanced if, for any two different points A, B in $\mathcal{S}$, there is a point C in $\mathcal{S}$ such that $AC = BC$. We say that $\mathcal{S}$ is centre-free if for any three points A, B, C in $\mathcal{S}$, there is no point P in $\mathcal{S}$ such that $PA = PB = PC$.
Show that for all integers $n \geq 3$, there exists a balanced set consisting of n points. Determine all integers $n \geq 3$ for which there exists a balanced centre-free set consisting of n points. (IMO 2015/1)
7.5: Let $\mathbb{Z}$ be the set of integers. Find all functions $f : \mathbb{Z} \to \mathbb{Z}$ such that

$$xf(2f(y) - x) + y^2 f(2x - f(y)) = \frac{f(x)^2}{x} + f(yf(y))$$

for all $x, y \in \mathbb{Z}$ with $x \neq 0$. (USAMO 2014/2)
8: For each positive integer n, the Bank of Cape Town issues coins of denomination $\frac{1}{n}$. Given a finite collection of such coins (of not necessarily different denominations) with total value at most most $99 + \frac{1}{2}$, prove that it is possible to split this collection into 100 or fewer groups, such that each group has total value at most 1. (IMO 2014/5)
8.5: Let I be the incentre of acute triangle ABC with $AB \neq AC$. The incircle ω of ABC is tangent to sides BC, CA, and AB at D, E, and F, respectively. The line through D perpendicular to EF meets ω at R. Line AR meets ω again at P. The circumcircles of triangle PCE and PBF meet again at Q.
Prove that lines DI and PQ meet on the line through A perpendicular to AI. (IMO 2019/6)
9: Let k be a positive integer and let S be a finite set of odd prime numbers. Prove that there is at most one way (up to rotation and reflection) to place the elements of S around the circle such that the product of any two neighbors is of the form $x^2 + x + k$ for some positive integer x. (IMO 2022/3)
9.5: An anti-Pascal triangle is an equilateral triangular array of numbers such that, except for the numbers in the bottom row, each number is the absolute value of the difference of the two numbers immediately below it. For example, the following is an anti-Pascal triangle with four rows which contains every integer from 1 to 10.

$$\begin{array}{ccccccc} & & & 4 & & & \\ & & 2 & & 6 & & \\ & 5 & & 7 & & 1 & \\ 8 & & 3 & & 10 & & 9 \end{array}$$

Does there exist an anti-Pascal triangle with 2018 rows which contains every integer from 1 to $1 + 2 + 3 + \cdots + 2018$? (IMO 2018/3)
10: Prove that there exists a positive constant c such that the following statement is true: Consider an integer $n > 1$, and a set $\mathcal{S}$ of n points in the plane such that the distance between any two different points in $\mathcal{S}$ is at least 1. It follows that there is a line ℓ separating $\mathcal{S}$ such that the distance from any point of $\mathcal{S}$ to ℓ is at least $cn^{-1/3}$.
(A line ℓ separates a set of points S if some segment joining two points in $\mathcal{S}$ crosses ℓ.) (IMO 2020/6)

 # Return format
Please return the corresponding difficulty scale (integer) in \box{}

Table 6: Prompt for difficulty estimation using LLM as a judge.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We proposed AdaRFT, a reinforcement finetuning paradigm integrated with curriculum learning to speedup the training.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discussed the limitation of our work in Section **??**. Specifically, we mentioned that ADARFT is less useful when the dataset is already well-balanced or the model is strong enough to handle problems with different difficulty levels.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: we provided a theoretical justification in Section 2 on why setting the target reward $\beta = 0.5$ may yield the best results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We detailed our implementation in Appendix E. We will also release our dataset and code after the anonymity period.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We submitted our code and dataset alongside the paper for reviewers.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://nips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer: [Yes]

Justification: we provide all implementation details in Section 2, 3, and Appendix E.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: we perform significance tests for the correlation of different difficulty metrics. We also show the raw as well as the smoothed trainning curves in 1.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).

- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: we provide all the information in Section 3.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: we use publicly available dataset, code, and training resources.

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [NA]

    Justification: this paper proposes ADARFT, a framework designed to accelerate RFT training. There are no specific societal impacts of this work beyond those already associated with RFT.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.
    - If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.

- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [NA]

    Justification: the paper poses no such ricks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: All the dataset and code we used are properly cited. We also did not violate the licenses of each asset.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

Answer: [Yes]

Justification: we submit our implementation of ADARFT alongside with the paper. See the readme file of our code for more details.

Guidelines:

- The answer NA means that the paper does not release new assets.
- Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
- The paper should discuss whether and how consent was obtained from people whose asset is used.
- At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

Answer: [NA]

Justification: No crowdsourcing and research with human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
- According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

Answer: [NA]

Justification: the paper does not involve crowdsourcing nor research wit human subjects.

Guidelines:

- The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does not impact the core methodology, scientific rigorousness, or originality of the research, declaration is not required.

    Answer: [NA]

    Justification: The use of LLMs is not an important, original, nor non-standard component of the core methods in this research.

    Guidelines:

    - The answer NA means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy (https://neurips.cc/Conferences/2025/LLM) for what should or should not be described.