GCPO: When Contrast Fails, Go Gold

Hao Wu*, **Wei Liu***
{howu, gyroliu}@tencent.com

***Abstract*—Reinforcement learning has been widely applied to enhance the reasoning capabilities of large language models. Extending the inference limits of smaller models has become a prominent research focus. However, algorithms such as Group Relative Policy Optimization (GRPO) suffer from a clear drawback: the upper bound of a model's rollout responses is entirely determined by the model itself, preventing the acquisition of knowledge from samples that are either all incorrect or all correct. In this paper, we introduce Group Contrastive Policy Optimazation (GCPO), a method that incorporates external standard reference answers. When the model cannot solve a problem, the reference answer supplies the correct response, steering the model toward an unequivocally accurate update direction. This approach offers two main advantages: (1) it improves training efficiency by fully utilizing every sample; (2) it enables the model to emulate the problem-solving strategy of the reference answer during training, thereby enhancing generalization in reasoning. GCPO achieves outstanding results across multiple benchmark datasets, yielding substantial improvements over the baseline model. Our code is available at: https://github.com/AchoWu/GCPO.**

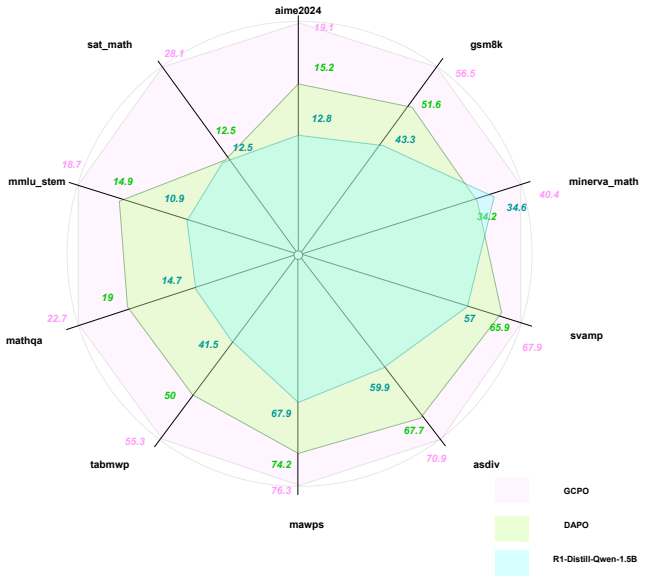

Figure 1: The performance of GCPO and DAPO on DeepSeek-R1-Distill-Qwen-1.5B across 10 math benchmarks.

1. Introduction

Reinforcement learning has gradually emerged as a novel paradigm for the post-training of large language models (LLMs) [1]. In particular, the OpenAI-o1 [2] and DeepSeek-R1 [3] series have further demonstrated the effectiveness of test-time scaling in boosting model inference capabilities [4, 5].

Existing literature [6, 7] characterises RLHF as a contrastive learning method. LLMs that conduct it through a reward maximization approach exhibit better generalization capabilities. As is well known, supervised learning only supplies "correct answers." It does not provide negative feedback that would enable models to differentiate between right and wrong responses, limiting improvements in reasoning ability [8, 9]. In this paper, we contend that GRPO [10] operates in a similar manner: it treats replies with higher intra-group advantage as positive examples and those with lower advantage as negative examples, thereby biasing the model's outputs toward the positives and away from the negatives. Severl works [11–14] have been designed to train the model to generate positive samples, yet the resulting outputs can never escape the model's inherent baseline performance. This observation raises the core question of our study: **must positive samples be generated by the training model**?

During the rollout phase, the generated responses may be either correct or incorrect. For questions that are intrinsically difficult, the model may never produce a correct answer, resulting in the absence of positive samples. Consequently, in the early stages of training many queries yield no correct responses, causing the intra-group advantage to be zero; similarly, in the later stages, when most samples are answered correctly, the intra-group advantage again collapses to zero. In both cases the policy gradient vanishes, which inflates the variance of the gradient estimate and hampers learning efficiency. Even if DAPO [11] filters out these samples, GRPO still cannot get the full use of training data. This observation motivates our second research question: **how can we enhance the utilisation of samples during training**?

Existing studies suffer from a fundamental misconception: they assume that simply presenting a problem alongside its standard answer enables the model to establish a clear link between the two [15]. In practice, small models often fabricate a plausible logical trajectory based on the supplied

*Equal contribution.

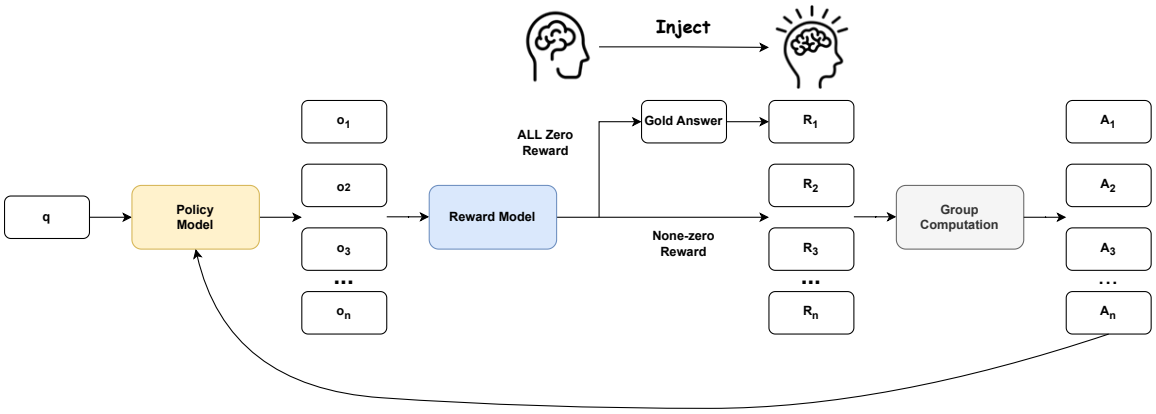

Figure 2: An overview of GCPO

answer, producing classic hallucinations. When a model repeatedly fails to answer a question, it indicates that the query lies beyond its capability [16], and the model may be unable to uncover any relationship between the question and the answer. Therefore, we argue that it is essential to introduce external guidance during training. By providing a standard answer that includes a complete and correct chain-of-thought, we can steer the model's updates toward the intended reasoning path. To address this, we draw inspiration from the use of "correct answers" in supervised learning and incorporated this idea into reinforcement learning.

Based the above observations, we propose a new variant algorithm: **G**roup **C**ontrastive **P**olicy **O**ptimazation (GCPO). We replace one of the responses with a gold-standard answer (golden answer, GA) when all rollouts fail, thereby supplying the model with a positive example that explicitly indicates the direction of optimization. This reference answer may be the true ground-truth or a response generated by a larger language model. Our goal is that, during training, the model gradually learns the reference answer and adopts the problem solving strategy of the larger model, thereby surpassing the baseline reasoning capability of the smaller model.

From a theoretical point of view, all rollouts for a single question are equivalent in weights. Consequently, any response within a rollout where all answers are incorrect can serve as a surrogate for the GA. Its position is not fixed, in this paper we choose to replace the first response. The overview of GCPO is presented in Fig. 2. By replacing the data produced by the model rollout data with a GA, we effectively supply the model with a positive example that indicates the correct reference direction in which it should be optimized. This is especially critical in situations where the model fails to answer the question correctly.

As described in GSPO [13], the Token-level importance sampling is unnecessary and may even harm model performance. In GRPO, rule-based rewards are computed based on the entire generated response sequence. This reward evaluates the performance of the entire sequence, rather than any individual token within it [14, 17]. Using sequence-level reward signals to drive token-level probability optimization constitutes a fundamental flaw in GRPO. By designing a sequence-level importance sampling, we can ensure the consistency between reward signals and optimization. In a nutshell, we make the following contributions:

- We introduce a novel reinforcement-learning algorithm that incorporates CoT data and supplies a reference answer to guide GRPO updates. This gives each update a clear direction, speeds convergence, and improves training sample efficiency.
- We show that, for reasoning tasks, KL divergence does not enhance training stability; instead, it limits model performance. Besides, we achieve alignment between reward signals and policy optimization by sequence-level token-level importance sampling.
- Extensive experiments demonstrate that our method achieves superior performance. It converges faster and enables a smaller model to emulate the reasoning style of a larger model.

2. Related Work

Chain of Thought Reasoning. Chain of Thought (CoT) [18] has been proposed to guide a model to generate a sequence of logical steps before producing an answer. By breaking complex tasks into manageable sub-problems, it arrives at the final solution [19]. This step-by-step problem-solving structure improves accuracy and enhances the model's ability to generalize [20]. Program-of-Thought [21], extends CoT by teaching models to use external tools—such as a Python interpreter—to solve math problems [22]. Moreover, the LLM can think self-consistency [23] or reflect, allowing it to correct mistakes and thereby improve accuracy. It has also been discovered that supervised fine-tuning on non-reasoning models using CoT dataset can elicit thier reasoning abilities [24, 25] or knowledge distillation [26, 27] from a larger teacher LLM [28–30]. However, as we discussed earlier, direct supervised fine-tuning may not necessarily provide the model

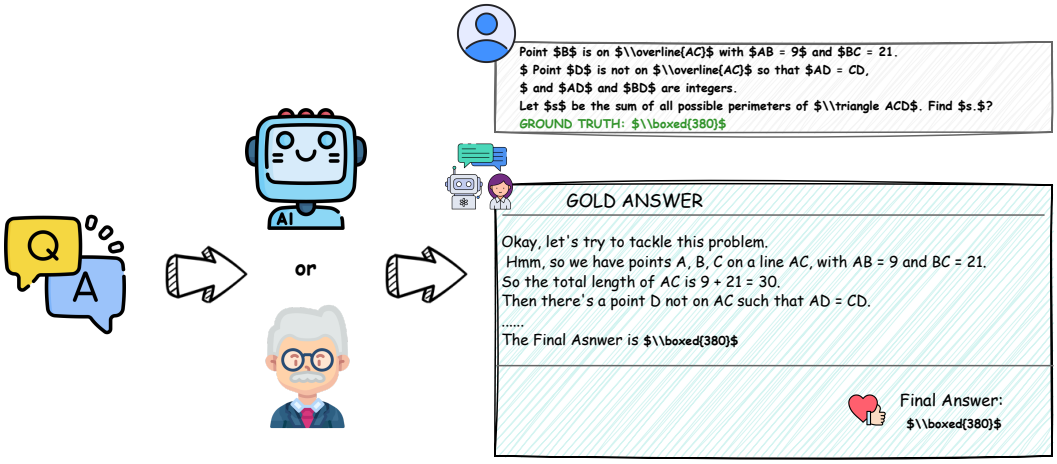

Figure 3: The illustration of GA

with sufficient information feedback to help it break through its own reasoning boundary capabilities.

Reinforcement Learning for LLMs Reasoning. Test time scaling via reinforcement learning has emerged as the primary post-training paradigm for instilling sophisticated reasoning capabilities in contemporary LLMs [14, 31, 32]. Central to this approach are Proximal Policy Optimization (PPO) [33] and DPO [34], the most notably GRPO [10], which achieves strong performance across mathematics, coding tasks by estimating advantages through in-group comparisons via Reinforcement Learning with Verifiable Rewards (RLVR) [35]. Building on this foundation, subsequent methods to address sample inefficiency [11, 13, 36, 37]. Parallel progress in reward engineering[35, 38–40] and data curationn[17, 41], like Open-Reasoner-Zero's curated preference corpus [42].

Existing methods often let the model roam freely, wasting many steps on ineffective updates. Consequently, it is difficult for the model to quickly reach its full potential. To address this, we designed an algorithm that guides every update toward a pre-specified optimal training direction, resulting in more stable and efficient training.

3. Method

3.1. Preliminaries

GPRO employs the average reward of multiple sampled responses to estimate a baseline, thereby eliminating the reward model and dramatically reducing RLHF resource consumption. Generally, we collect a group of responses $\{o_i\}_{i=1}^{G}$ for a single question from the old policy model $\pi_{\theta_{old}}$ and calculate the advantages $A_{i,t}$ through the rewards of parsed responses. The surrogate objective is defined as follows:

$$\mathcal{L}_{\text{GRPO}}(\theta) = -\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|} \left[\min\left(r(\theta)\hat{A}_{i,t},\ \text{clip}\left(r_{i,t}(\theta), 1-\epsilon, 1+\epsilon\right)\hat{A}_{i,t}\right)\right] \tag{1}$$

where

$$r_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t})};\ \hat{A}_{i,t} = \frac{R_i - \text{mean}(\{R_i\}_{i=1}^{G})}{\text{std}(\{R_i\}_{i=1}^{G})} \tag{2}$$

$r(\theta)$ is the important sampling ratio. This relative formulation encourages the model to prefer the positive response within each group.

3.2. GCPO

It is generally accepted that responses with higher advantage are correct and thus indicate the direction for model updates. However, when a question is highly complex or difficult, the model's rollouts may contain no correct samples; the baseline then offers no guidance and the model lacks an update direction. In DAPO, problems that are either extremely simple or extremely difficult (all rewards for the question are zero) are typically discarded. This practice not only reduces data utilization efficiency, but also, as argued in this paper, results in the loss of difficult samples—samples that are crucial for enhancing the model's reasoning capabilities.

Motivated by this observation, we hypothesize that every question has a golden answer (GA) — the correct reference answer. Using this GA as the update target should consistently improve training outcomes as show in Fig. 3. It is reasonable for mathematics and coding problems, only questions with correct solutions are included in training and test sets. We can generate responses for a given problem with a larger LLM or have humans annotate them. Meanwhile, integrating these standard answers into the training process

enables the model to learn the thinking patterns for solving problems, thereby improving its performance.

$$\hat{A}_i = \begin{cases} \frac{R_i - \text{mean}(\{R_i\}_{i=1}^G)}{\text{std}(\{R_i\}_{i=1}^G)} & \text{if } \mathbf{R} \neq \mathbf{0}. \\ \frac{R'_i - \text{mean}(\{R'_i\}_{i=1}^G)}{\text{std}(\{R'_i\}_{i=1}^G)} & \text{if } \mathbf{R} = \mathbf{0}. \end{cases} \tag{3}$$

Here $\mathbf{R} = \mathbf{0}$ indicates the degenerate case when every rollout reward is zero. If at least one reward is nonzero ($\mathbf{R} \neq \mathbf{0}$), the advantage is computed by normalizing the original reward $\mathbf{R}$. When all rewards are zero, we instead normalize the new reward $\mathbf{R}'$ obtained by substituting one failed rollout response with the golden answer (so that $R'_{\text{GoldAnswer}} = 1$ and $R'_{\text{else}} = 0$ in Eq. (3)).

Furthermore, the vanilla GPRO's importance sampling is applied at the token-level. The rule-based rewards are computed based on the entire generated response sequence. For example, when addressing a math problem, the model generates a text sequence containing multiple reasoning steps, culminating in the final answer. The rule-based verifier checks whether the final answer is correct and then assigns a reward (e.g., 1 for correct, 0 for incorrect). This reward only evaluates the performance of the entire sequence, not individual tokens within it.

Extensive discussion and empirical evidence, however, demonstrated that this granularity is suboptimal. GSPO [13] addresses the issue by unifying importance sampling to the sequence level. Following this insight, we changed the token-level importance sampling to sequence-level to further enhance training stability, as Eq. (4). Subsequent experiments demonstrate that this minor modification not only yields improvements in performance but also enhances the stability of training.

$$r_i(\theta) = \frac{\pi_\theta(o_i \mid q)}{\pi_{\text{old}}(o_i \mid q)} \tag{4}$$

3.3. Other techniques

As in DAPO [11], we also omit the KL divergence penalty from our objective. In CoT, extensive reasoning—encompassing reflection and retry—tends to cause the output distribution to deviate markedly from that of the base model. Imposing a KL constraint would thus impede these updates, a consequence we consider detrimental to training. Consequently, the KL divergence term is excluded from our loss function.

In the end, our total objective function is shown in Eq. (5).

$$\mathcal{L}_{\text{OURS}}(\theta) = -\frac{1}{G}\sum_{i=1}^{G} \text{clip}\left(r_i(\theta), 1-\epsilon_{\text{low}}, 1+\epsilon_{\text{high}}\right)\hat{A}_i \tag{5}$$

4. Experiments

4.1. Experiment Setup

We conduct our experiments on DeepSeek-R1-Distill-Qwen-1.5B(abbreviated as R1-1.5B) [3]. We did not select the latest Qwen-3 Series [43] models primarily due to concerns about issues such as data leakage, which could lead to abnormal model performance. We select 10 benchmarks for evalution, including: AIME2024 [44], GSM8K [45], MATH [46], ASDiv [47], MAWPS [48], TAB(short for TabMWP) [49], MQA(short for MathQA) [50], MMLU (only evaluate on STEM subjects) [51], SAT(SAT math). For Evaluation metrics, we use accuracy (pass@1) as the evaluation metrics by default, and select mean@32 when evaluating AIME2024 dataset.

We use verl [52] as the training framework, and all experiments are running on 8 H20 GPUs. The rollout number during training is fixed to 16. The temperature is set to 0.7 for training and inference. During the training process, we monitor the rewards of rollouts. When there is a rollout where all outcomes are incorrect, GCPO will use GA to replace the corresponding response. The other training parameters are followed by the default settings of DAPO.

For the training data, we employ the widely used open source collection [17], namely DAPO-Math-17k-Processed[1] , which has been reformatted from the original DAPO dataset. When constructing the Golden Answer (GA) set, we follow the standard practice of using labeled data directly for typical tasks, as in supervised training. However, our objective is to endow the reinforcement-learned model with a stronger reasoning capability. Therefore, GA is defined as a set of canonical answers—analogous to those used in mathematics. To generate these canonical solutions, we employ a more powerful model, DeepSeek-R1 [3], to produce responses to each prompt. This approach serves two purposes: (1) Filtering out intractable problems. By discarding instances for which the target model fails to produce a correct solution, we eliminate overly difficult problems that would otherwise provide no useful training signal. Simultaneously, the generated chain-of-thought (CoT) process allows a smaller model to acquire the larger model's reasoning patterns, particularly benefiting models with limited baseline reasoning skills. (2) Enforcing token and format constraints. Due to the maximum-token limit in our settings, any response exceeding the predefined threshold is excluded. Additionally, samples whose response format does not conform to the specified standards are removed. After these filtering steps, the resulting training set contains 9,975 samples.

4.2. Main Results

This section evaluates our method against DAPO under identical experimental conditions, quantifying the performance gaps relative to the baseline model R1-1.5B. Figure Fig. 1 shows that our approach consistently outperforms both DAPO and the baseline across almost all datasets. On the AIME 2024 dataset, GCPO achieves a 25% improvement over DAPO at the mean@32. Compared with the baseline model R1-1.5B, GCPO delivers roughly a 54% performance gain on the dataset MQA. This substantial boost generalises

1. https://huggingface.co/datasets/open-r1/DAPO-Math-17k-Processed

TABLE 1: The ACC (%) of R1-1.5B under different training settings. TIS stands for Token-level Important Sampling.

	gsm8k	minerva_math	tabmwp	mathqa	mmlu_stem	sat_math	Average
R1-1.5B	43.3	34.6	41.5	14.7	10.9	12.5	26.25
DAPO	51.6	34.2	50.0	19.0	14.9	12.5	30.37
DAPO w/o TIS	50.9	37.4	49.5	17.3	15.2	25	32.55
GCPO w/ KL	47.2	31.8	46.4	**27.5**	**20.2**	25	33.02
GCPO w/ TIS	42.8	31.8	40.9	26.6	17.5	**34.4**	32.33
GCPO	**56.5**	**40.4**	**55.3**	22.7	18.7	28.1	**36.95**

to nearly all evaluation sets, indicating the robustness of the proposed method.

4.3. Ablation Study

In this section we examine the functions of the different modules within GCPO. Tab. 1 lists the baseline model (R1-1.5B), DAPO, DAPO w/o TIS (Token-level Importance Sampling removed), GCPO w/ KL (including KL divergence), GCPO w/ TIS (utilizing Token-level Importance Sampling), and the full GCPO. Across six evaluation benchmark datasets, GCPO achieves the best performance at 36.95%, markedly surpassing DAPO's 30.37%.

Token-level and Sequence-level Important Sampling. Through comparisons across multiple experiments, we find that token-level importance sampling exerts a negative impact on model performance, with the average performance dropping from 36.95% to 32.33%. These results are consistent with the conclusions drawn in prior work [11, 13, 37].

Removing KL Divergence. From the result of GCPO w/ KL, we observe that adding a KL divergence penalty actually degrades performance, a phenomenon also was mentioned in DAPO. After reinforcement training, the model's distribution diverges substantially from the original baseline model distribution. Penalizing this shift with KL constrains the updates. For training large language models, training stability is paramount, overly small update steps can trap the model in local optima and hinder its ability to surpass its existing capability boundaries.

In summary, GCPO delivers a clear performance advantage over DAPO, particularly in reasoning-heavy benchmarks, thereby validating the effectiveness of its proposed policy-optimization scheme.

5. Conclusion

We introduce a novel method, GCPO. When a model's rollout yields only incorrect outputs, GCPO supplies a standard answer to explicitly steer the model toward the desired update at that step. When the model cannot solve a problem on its own, GCPO introduces GA to supply external knowledge. This gives the model a new, trustworthy direction to update and expands its reasoning capabilities. Among these, the introduction of sequence-level importance sampling and the removal of KL divergence contribute to more stable training and a more efficient performance. This approach not only boosts training efficiency but also enables smaller models to learn the reasoning patterns of larger models and incorporate them into their responses. Experiments show that this training method achieves outstanding performance. We hope GCPO will inspire new ideas in the field of model inference and anticipate that it will lead to fundamental advances in training larger-scale models.

6. Limitations

Our method first requires collecting the GAs (for the training set data) before it can be put into use, which consumes a certain amount of resources, such as calls to LLM or manual annotation. Secondly, our experimental evaluation is limited to mathematical tasks. Nonetheless, we contend that GCPO possesses far broader applicability. For example, it can be integrated with tool-using CoT frameworks to train models that invoke external tools across a wide spectrum of problems. We hope that this work will motivate the development of more efficient strategies for augmenting models' logical reasoning capabilities.

References

[1] Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744, 2022.

[2] Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

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

[4] Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? In *2nd AI for Math Workshop @ ICML 2025*, 2025.

[5] Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

[6] Xufei Lv, Haoyuan Sun, Xuefeng Bai, Min Zhang, Houde Liu, and Kehai Chen. The hidden link between rlhf and contrastive learning. *arXiv preprint arXiv:2506.22578*, 2025.

[7] Youssef Mroueh. Reinforcement learning with verifiable rewards: Grpo's effective loss, dynamics, and success amplification. *arXiv preprint arXiv:2503.06639*, 2025.

[8] Avinash Patil and Aryan Jadon. Advancing reasoning in large language models: Promising methods and approaches. *arXiv preprint arXiv:2502.03671*, 2025.

[9] Wei Shen, Xiaoying Zhang, Yuanshun Yao, Rui Zheng, Hongyi Guo, and Yang Liu. Improving reinforcement learning from human feedback using contrastive rewards. *arXiv preprint arXiv:2403.07708*, 2024.

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

[11] Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, et al. Dapo: An open-source llm reinforcement learning system at scale. *arXiv preprint arXiv:2503.14476*, 2025.

[12] Andre He, Daniel Fried, and Sean Welleck. Rewarding the unlikely: Lifting grpo beyond distribution sharpening. *arXiv preprint arXiv:2506.02355*, 2025.

[13] Chujie Zheng, Shixuan Liu, Mingze Li, Xiong-Hui Chen, Bowen Yu, Chang Gao, Kai Dang, Yuqiong Liu, Rui Men, An Yang, et al. Group sequence policy optimization. *arXiv preprint arXiv:2507.18071*, 2025.

[14] Weihao Zeng, Yuzhen Huang, Qian Liu, Wei Liu, Keqing He, Zejun MA, and Junxian He. SimpleRL-zoo: Investigating and taming zero reinforcement learning for open base models in the wild. In *Second Conference on Language Modeling*, 2025.

[15] Qihan Huang, Weilong Dai, Jinlong Liu, Wanggui He, Hao Jiang, Mingli Song, Jingyuan Chen, Chang Yao, and Jie Song. Boosting mllm reasoning with text-debiased hint-grpo. *arXiv preprint arXiv:2503.23905*, 2025.

[16] Qineng Wang, Zihao Wang, Ying Su, Hanghang Tong, and Yangqiu Song. Rethinking the bounds of LLM reasoning: Are multi-agent discussions the key? In Lun-Wei Ku, Andre Martins, and Vivek Srikumar, editors, *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6106–6131, Bangkok, Thailand, August 2024. Association for Computational Linguistics.

[17] Hugging Face. Open r1: A fully open reproduction of deepseek-r1, January 2025.

[18] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837, 2022.

[19] Yao Fu, Hao Peng, Ashish Sabharwal, Peter Clark, and Tushar Khot. Complexity-based prompting for multi-step reasoning. In *The Eleventh International Conference on Learning Representations*, 2023.

[20] Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA, 2022. Curran Associates Inc.

[21] Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *Transactions on Machine Learning Research*, 2023.

[22] Tatsuro Inaba, Hirokazu Kiyomaru, Fei Cheng, and Sadao Kurohashi. MultiTool-CoT: GPT-3 can use multiple external tools with chain of thought prompting. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1522–1532, Toronto, Canada, July 2023. Association for Computational Linguistics.

[23] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*, 2023.

[24] 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. *arXiv preprint arXiv:2501.19393*, 2025.

[25] Qiyuan Zhang, Fuyuan Lyu, Zexu Sun, Lei Wang, Weixu Zhang, Wenyue Hua, Haolun Wu, Zhihan Guo, Yufei Wang, Niklas Muennighoff, et al. A survey on test-time scaling in large language models: What, how, where, and how well? *arXiv preprint arXiv:2503.24235*, 2025.

[26] Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*, 2015.

[27] Jianping Gou, Baosheng Yu, Stephen J Maybank, and Dacheng Tao. Knowledge distillation: A survey. *International journal of computer vision*, 129(6):1789–1819, 2021.

[28] Namgyu Ho, Laura Schmid, and Se-Young Yun. Large language models are reasoning teachers. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14852–14882, Toronto, Canada, July 2023. Association for Computational Linguistics.

[29] Lucie Charlotte Magister, Jonathan Mallinson, Jakub Adamek, Eric Malmi, and Aliaksei Severyn. Teaching small language models to reason. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 1773–1781, Toronto, Canada, July 2023. Association for Computational Linguistics.

[30] Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. STar: Bootstrapping reasoning with reasoning. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[31] Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Li Erran Li, Raluca Ada Popa, and Ion Stoica. Deepscaler: Surpassing o1-preview with a 1.5b model by scaling rl. https://pretty-radio-b75.notion.site/DeepScaleR-Surpassing-O1-Preview-with-a-1-5B-Model-by-Scaling-RL-19681902c1468005bed8ca303013a4e2, 2025. Notion Blog.

[32] Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tulu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

[33] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[34] 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. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023.

[35] Yiping Wang, Qing Yang, Zhiyuan Zeng, Liliang Ren, Liyuan Liu, Baolin Peng, Hao Cheng, Xuehai He, Kuan Wang, Jianfeng Gao, et al. Reinforcement learning for reasoning in large language models with one training example. *arXiv preprint arXiv:2504.20571*, 2025.

[36] Zichen Liu, Changyu Chen, Wenjun Li, Penghui Qi, Tianyu Pang, Chao Du, Wee Sun Lee, and Min Lin. Understanding r1-zero-like training: A critical perspective. *arXiv preprint arXiv:2503.20783*, 2025.

[37] Xiaojiang Zhang, Jinghui Wang, Zifei Cheng, Wenhao Zhuang, Zheng Lin, Minglei Zhang, Shaojie Wang, Yinghan Cui, Chao Wang, Junyi Peng, et al. Srpo: A cross-domain implementation of large-scale reinforcement learning on llm. *arXiv preprint arXiv:2504.14286*, 2025.

[38] Shyam Sundhar Ramesh, Yifan Hu, Iason Chaimalas, Viraj Mehta, Pier Giuseppe Sessa, Haitham Bou Ammar, and Ilija Bogunovic. Group robust preference optimization in reward-free rlhf. *Advances in Neural Information Processing Systems*, 37:37100–37137, 2024.

[39] Rulin Shao, Shuyue Stella Li, Rui Xin, Scott Geng, Yiping Wang, Sewoong Oh, Simon Shaolei Du, Nathan Lambert, Sewon Min, Ranjay Krishna, et al. Spurious rewards: Rethinking training signals in rlvr. *arXiv preprint arXiv:2506.10947*, 2025.

[40] Mihir Prabhudesai, Lili Chen, Alex Ippoliti, Katerina Fragkiadaki, Hao Liu, and Deepak Pathak. Maximizing confidence alone improves reasoning. *arXiv preprint arXiv:2505.22660*, 2025.

[41] Lifan Yuan, Ganqu Cui, Hanbin Wang, Ning Ding, Xingyao Wang, Boji Shan, Zeyuan Liu, Jia Deng, Huimin Chen, Ruobing Xie, Yankai Lin, Zhenghao Liu, Bowen Zhou, Hao Peng, Zhiyuan Liu, and Maosong Sun. Advancing LLM reasoning generalists with preference trees. In *The Thirteenth International Conference on Learning Representations*, 2025.

[42] 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. *arXiv preprint arXiv:2503.24290*, 2025.

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

[44] AIME. Aime. aime problems and solutions, 2024., 2024.

[45] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

[46] Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, et al. Solving quantitative reasoning problems with language models. *Advances in neural information processing systems*, 35:3843–3857, 2022.

[47] Shen-yun Miao, Chao-Chun Liang, and Keh-Yih Su. A diverse corpus for evaluating and developing English math word problem solvers. In Dan Jurafsky, Joyce

Chai, Natalie Schluter, and Joel Tetreault, editors, *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 975–984, Online, July 2020. Association for Computational Linguistics.

[48] Rik Koncel-Kedziorski, Subhro Roy, Aida Amini, Nate Kushman, and Hannaneh Hajishirzi. MAWPS: A math word problem repository. In Kevin Knight, Ani Nenkova, and Owen Rambow, editors, *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1152–1157, San Diego, California, June 2016. Association for Computational Linguistics.

[49] Pan Lu, Liang Qiu, Kai-Wei Chang, Ying Nian Wu, Song-Chun Zhu, Tanmay Rajpurohit, Peter Clark, and Ashwin Kalyan. Dynamic prompt learning via policy gradient for semi-structured mathematical reasoning. In *The Eleventh International Conference on Learning Representations*, 2023.

[50] Aida Amini, Saadia Gabriel, Peter Lin, Rik Koncel-Kedziorski, Yejin Choi, and Hannaneh Hajishirzi. Mathqa: Towards interpretable math word problem solving with operation-based formalisms. *arXiv preprint arXiv:1905.13319*, 2019.

[51] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021.

[52] 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.