Multi-layer GRPO

Fei Ding[1]

Abstract

The success of DeepSeek-R1 (DeepSeek-AI et al., 2025) has demonstrated the effectiveness of the GRPO algorithm. However, due to the absence of process rewards, GRPO often suffers from inefficiencies in exploration, as a single detailed error can result in an entirely incorrect final answer, leading to zero rewards.To address these challenges, we propose MGRPO (Multi-layer GRPO). In the first layer, GRPO operates identically to the original version, generating an initial response. This response is then fed into a second-stage GRPO process, which primarily trains the model to correct errors. Experimental results indicate that MGRPO outperforms standard GRPO, achieving superior performance.

1. Introduction

Sparse Outcome Rewards provide feedback only at the final step of reasoning, which often leads to inefficiencies in reinforcement learning (RL) for large language models (LLMs). A minor mistake in an intermediate step can result in a completely incorrect final answer, leading to zero rewards and making the exploration process highly inefficient (Uesato et al., 2022; Lightman et al., 2023a; Wang et al., 2023; Yuan et al., 2024a).

In contrast, Dense Process Rewards demonstrate significant advantages in terms of reasoning scalability, training efficiency (Sutton & Barto, 2018), and credit assignment (Leike et al., 2018). However, the practical application of dense rewards in RL training remains limited (Setlur et al., 2024), and current state-of-the-art industrial models still primarily rely on verifiable outcome rewards without achieving breakthroughs in dense reward utilization (DeepSeek-AI et al., 2025; Team et al., 2025). The core challenge lies in how to efficiently obtain and leverage high-quality dense rewards and integrate them effectively into policy learning (Gao et al., 2022).

To address these challenges, we propose MGRPO (Multi-layer GRPO), which unifies the generative model and process reward model into a single framework. For each data instance, MGRPO simultaneously trains the model in both aspects. It employs a multi-layer Group Relative Policy Optimization (GRPO) approach, where the data generated in the first GRPO layer is used to synthesize labeled data for training the process reward model in the second GRPO layer, enabling a spiral improvement of the model's capabilities. MGRPO introduces a two-layer GRPO process:

- First Layer - Standard GRPO: The model generates initial responses using GRPO, following the conventional approach.

- Second Layer: The input and output from the first stage are concatenated, followed by some guiding prompts as input, guiding the model to identify and correct errors. If errors cannot be corrected, the data is discarded. If successfully revised, GRPO training is applied to reinforce the model's ability to rectify mistakes.This process implicitly includes a reward for the first layer GRPO process.

Experimental results demonstrate that MGRPO outperforms standard GRPO in mathematical reasoning tasks. Specifically, in the Qwen2.5-Math-7B-Base benchmark, MGRPO achieves superior performance compared to using GRPO alone. MGRPO effectively replaces the costly human-annotated process rewards by reusing data generated through the GRPO process (Yuan et al., 2024a).We found that as the error correction ability improves, the first-layer generation also becomes more capable of producing correct answers.

Overall, MGRPO introduces a structured multi-layer learning paradigm that enables LLMs to simultaneously enhance their reasoning capabilities and error correction abilities, allowing the model to learn from its own mistakes. Furthermore, MGRPO can be generalized to other RL algorithms, opening new directions for reinforcement learning in large-scale models.

---

[1]. Correspondence to: Fei Ding <dingfei@email.ncu.edu.cn>.

Proceedings of the 41st International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024. Copyright 2024 by the author(s).

2. Background and Related Work

Following the success of DeepSeek-R1 (DeepSeek-AI et al., 2025), rule-based reinforcement learning has garnered significant attention. DeepSeek-AI et al. (2025) observed that self-correction capability naturally emerges during reinforcement learning training, a phenomenon they referred to as the ”AHA moment” in their study.Open-source replication studies based on Qwen-2.5-Math (Liu et al., 2025; Zhang et al., 2025; Cheng et al., 2025) indicate that while the base model exhibits a certain degree of self-correction ability, this capability remains sparse. Through our replication of DeepSeek-R1 and continuous manual inspection of every model output, we identified the underlying mechanism behind the emergence of the AHA moment: (1) The pretraining corpus inherently contains self-correction patterns, which endow the model with an intrinsic but latent self-correction ability. (2) During inference, self-correction behaviors can occur probabilistically. When a self-correction instance happens to yield the correct answer, it receives a positive reward, thereby reinforcing the likelihood of self-correction behaviors through iterative learning.

(Yuan et al., 2024b) propose integrating the reward model into the language model itself, enabling it to generate and evaluate training data autonomously, thereby achieving self-alignment. However, this approach poses a risk of reward hacking, where the model inflates rewards by generating excessively verbose responses that are not necessarily of higher quality. Additionally, its improvements in mathematical and logical reasoning tasks remain limited. Our MGRPO method relies on an external ground-truth reward model to provide training signals, effectively mitigating the risk of reward hacking.

Li et al. (2024) propose a method where a teacher model generates multiple reflective responses, and a student model selects the most compatible one for supervised fine-tuning (SFT). This approach enhances the student model’s self-reflection capability.

(Kumar et al., 2024) proposes a reinforcement learning method called SCoRe (Self-Correction via Reinforcement Learning), designed to train large language models (LLMs) to achieve effective self-correction using self-generated data. Through multiple rounds of online reinforcement learning, SCoRe addresses two key issues in traditional methods (such as Supervised Fine-Tuning, SFT): distribution mismatch (where training data does not align with the distribution of the model’s own errors) and behavioral collapse (where the model tends to generate an optimal initial response, making subsequent corrections ineffective). Experiments demonstrate that SCoRe significantly outperforms baseline models in mathematical reasoning (MATH) and code generation (HumanEval) tasks, achieving self-correction performance improvements of 15.6% and 9.1%, respectively. SCoRe suffers from the issue of behavioral collapse, which requires the introduction of two-stage RL training, reward shaping, and other methods, thereby increasing the complexity of the training process. In contrast, our MGRPO method avoids the problem of behavioral collapse through careful design.

3. Methods

3.1. GRPO

We employ Group Relative Policy Optimization (GRPO), which eliminates the necessity for an additional value function approximation as required in Proximal Policy Optimization (PPO). Instead, GRPO utilizes the average reward of multiple sampled outputs—generated in response to the same query—as a baseline. Formally, given a query q, GRPO first samples a set of responses $\{o_1, o_2, \cdots, o_G\}$ from the previous policy $\pi_{\theta_{old}}$. The policy is then updated by optimizing the following objective:

$$\begin{aligned}\mathcal{J}_{GRPO}(\theta) = \mathbb{E}[q \sim P(Q), \{o_i\}_{i=1}^{G} \sim \pi_{\theta_{old}}(O|q)]&\\ \frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\{\min[\frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q,o_{i,<t})}\hat{A}_{i,t},&\\ \text{clip}\left(\frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q,o_{i,<t})}, 1-\epsilon, 1+\epsilon\right)\hat{A}_{i,t}]&\\ -\beta\mathbb{D}_{KL}\left[\pi_\theta||\pi_{ref}\right]\},&\end{aligned} \tag{1}$$

where ϵ and β are hyper-parameters, and $\hat{A}_{i,t}$ represents the advantage computed exclusively based on the relative rewards within each sampled group. GRPO’s group-relative approach to advantage estimation is inherently aligned with the comparative nature of reward models, as these models are typically trained on datasets comprising pairwise comparisons of outputs for the same query. Notably, instead of incorporating a KL penalty term directly into the reward, GRPO regularizes the policy by explicitly adding the KL divergence between the trained policy and a reference policy to the loss function. This design choice simplifies the computation of $\hat{A}_{i,t}$ by avoiding additional complexity introduced by KL-based reward penalties. Furthermore, in contrast to the KL penalty used in Proximal Policy Optimization (PPO), GRPO estimates the KL divergence using the following unbiased estimator, as proposed in Schulman (2020):

$$\mathbb{D}_{KL}\left[\pi_\theta||\pi_{ref}\right] = \frac{\pi_{ref}(o_{i,t}|q,o_{i,<t})}{\pi_\theta(o_{i,t}|q,o_{i,<t})} - \log\frac{\pi_{ref}(o_{i,t}|q,o_{i,<t})}{\pi_\theta(o_{i,t}|q,o_{i,<t})} - 1, \tag{2}$$

which is guaranteed to be positive.

4. MGRPO: Multi-layer Group Relative Policy Optimization

MGRPO extends GRPO by introducing a multi-layer structure, leveraging a generate-correct-select-augment process to enhance model optimization.

4.1. Layer1-GRPO

The first layer follows the standard GRPO framework, where group-wise sampling and relative advantage estimation are applied:

$$\begin{aligned}\mathcal{J}_{\text{Layer1-GRPO}}(\theta) = &\mathbb{E}_{q\sim P(Q),\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{old}}(O|q)}\Bigg[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\\ &\min\Big(r_{i,t}(\theta)\hat{A}_{i,t},\\ &\text{clip}(r_{i,t}(\theta),1-\epsilon,1+\epsilon)\hat{A}_{i,t}\Big)\\ &-\beta\mathbb{D}_{KL}(\pi_\theta||\pi_{ref})\Bigg],\end{aligned} \tag{3}$$

where:

- $r_{i,t}(\theta)=\frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q,o_{i,<t})}$ is the importance ratio.

- $\hat{A}_{i,t}$ represents the relative advantage estimated solely within each sampled group.

- The KL divergence term $\mathbb{D}_{KL}(\pi_\theta||\pi_{ref})$ constrains policy deviation.

4.2. Layer2-GRPO

In the second layer, MGRPO introduces a correction-selection-augmentation mechanism:

1. **Correction**: Each response o_i is transformed into a new query q_i':

   $$q_i' = f_{\text{concat}}(q,o_i)+p_{\text{guide}}, \tag{4}$$

   where f_{concat} concatenates the original query with the output, and p_{guide} is a randomly selected guiding phrase.

   The guiding prompts are randomly selected from a predefined pool to prevent the model from overfitting to a single template. Examples of guiding prompts p_{guide} are as follows:

   - Where might I have gone wrong this time? Let me double-check carefully.
   - Wait, let me double-check that.
   - Wait a minute, let me make sure I didn't make a mistake.
   - Hmm, let me think if there's another way to approach this problem.
   - Wait, maybe I can think about it like this:
   - Another thought: maybe I can
   - But wait, let me just make sure I didn't miss anything in the original problem.

2. **Selection**: A rule-based verifier filters valid corrected responses $\{\tilde{o}_j\}_{j=1}^{G'}$ (where $G'\leq G$).

3. **Augmentation**: Each selected response $\tilde{o}_j$ is augmented by generating H additional variations:

   $$\{\tilde{o}_j^{(1)},\tilde{o}_j^{(2)},\ldots,\tilde{o}_j^{(H)}\}. \tag{5}$$

The Layer2-GRPO objective follows the same form as Layer1-GRPO:

$$\begin{aligned}\mathcal{J}_{\text{Layer2-GRPO}}(\theta) = &\mathbb{E}_{q'\sim P(Q'),\{\tilde{o}_j^{(h)}\}_{h=1}^{H}\sim\pi_{\theta_{old}}(O|q')}\\ &\Bigg[\frac{1}{H}\sum_{j=1}^{G'}\sum_{h=1}^{H}\frac{1}{|\tilde{o}_j^{(h)}|}\sum_{t=1}^{|\tilde{o}_j^{(h)}|}\\ &\min\Big(\tilde{r}_{j,t}^{(h)}(\theta)\hat{A}_{j,t}^{(h)},\\ &\text{clip}(\tilde{r}_{j,t}^{(h)}(\theta),1-\epsilon,1+\epsilon)\hat{A}_{j,t}^{(h)}\Big)\\ &-\beta\mathbb{D}_{KL}(\pi_\theta||\pi_{ref})\Bigg].\end{aligned} \tag{6}$$

where:

- q' is the modified query after correction.

- $\tilde{o}_j^{(h)}$ represents the corrected, selected, and augmented responses.

- $\tilde{r}_{j,t}^{(h)}(\theta)$ and $\hat{A}_{j,t}^{(h)}$ are the importance ratio and relative advantage estimates for Layer2.

5. Experiment

Tasks. We mainly focus on reasoning problems in math : **(1) MATH** (Hendrycks et al., 2021): following (Lightman et al., 2023b), we augment the MATH training set with 4500 problems from the test set, and report results on the remaining 500 problems (MATH500).

(2) gsm8k (Cobbe et al., 2021):We report results on 1000 test samples.

(3) Minerva Math (Lewkowycz et al., 2022) is primarily used to assess the capability of language models in quantitative reasoning tasks. It consists of scientific and mathematical problems that require multi-step reasoning.

(4) OlympiadBench (He et al., 2024) consists of 8,476 math and physics problems from the International Olympiads,

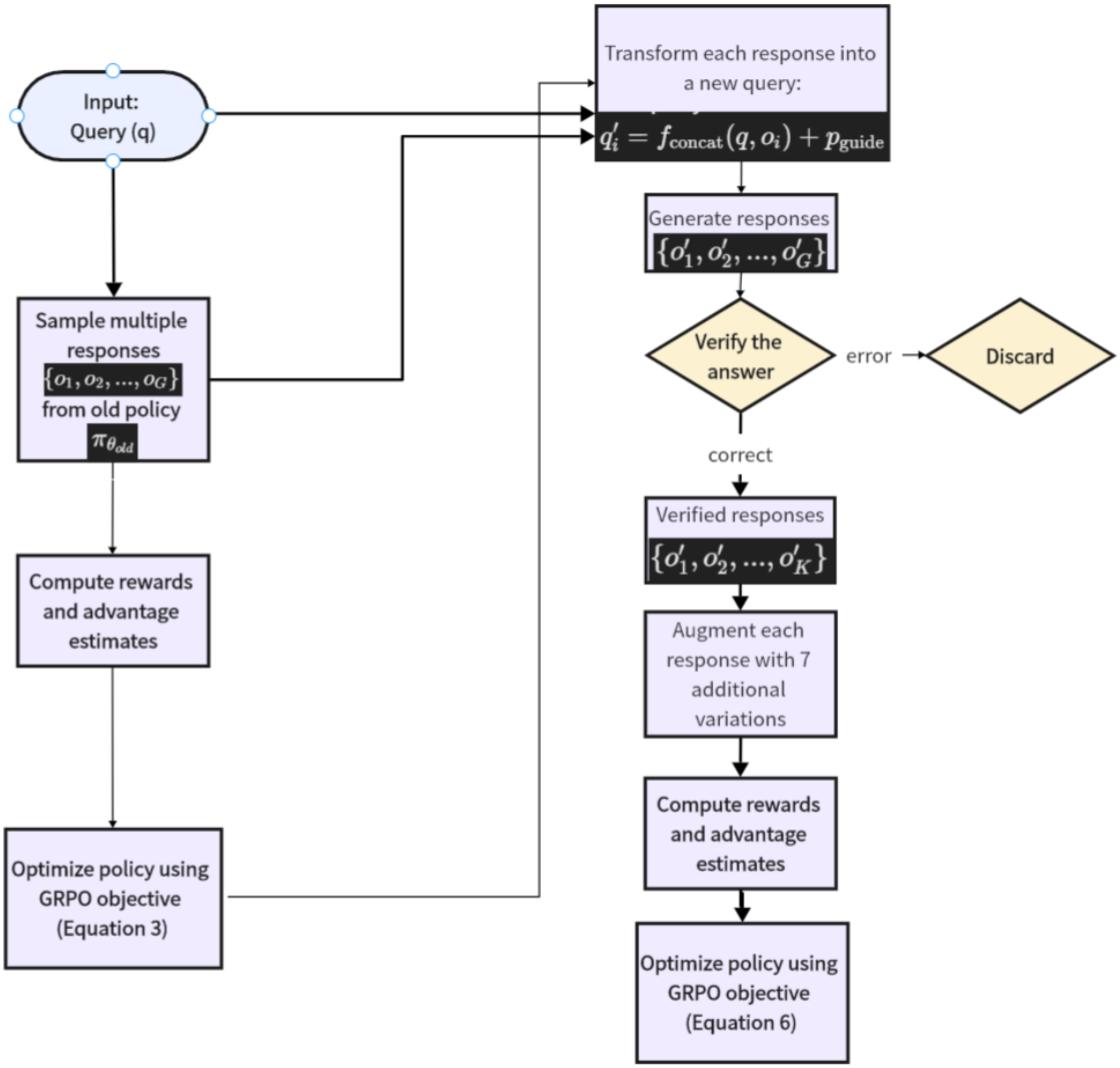
Input:
Query (q)
Sample multiple responses
$\{o_1, o_2, ..., o_G\}$
from old policy
$\pi_{\theta_{old}}$
Compute rewards and advantage estimates
Optimize policy using GRPO objective (Equation 3)
Transform each response into a new query:
$q'_i = f_{\text{concat}}(q, o_i) + p_{\text{guide}}$
Generate responses
$\{o'_1, o'_2, ..., o'_G\}$
Verify the answer
error
Discard
correct
Verified responses
$\{o'_1, o'_2, ..., o'_K\}$
Augment each response with 7 additional variations
Compute rewards and advantage estimates
Optimize policy using GRPO objective (Equation 6)

Figure 1. Flowchart of MGRPO.

Layer1 grpo:

Input:
```
<|begin-of-sentence|><|User|>Tobias is buying a new pair of shoes that
costs $95.  He has been saving up his money each month for the past three
months.  He gets a $5 allowance a month.  He also mows lawns and shovels
driveways.  He charges $15 to mow a lawn and $7 to shovel.  After buying the
shoes, he has $15 in change.  If he mows 4 lawns, how many driveways did he
shovel?<|Assistant|>
```

output:
```
<think> First, I need to determine how much money Tobias has saved in
total.  The shoes cost $95, and after purchasing them, he has $15 left.  This
means he had $110 in total savings.  , I'll calculate how much he earned from
mowing lawns.  He mowed 4 lawns at ...Finally, I'll verify that 7 driveways
yield $49, leaving exactly $1 left over, which matches the given information.
</think> Let's break down the problem step by step to find out how ...
```

Layer2 grpo:

Input:
```
<|begin-of-sentence|><|User|>Tobias is buying a new pair of shoes that
costs $95.  He has been saving up his money each month for the past three
months.  He gets a $5 allowance a month.  He also mows lawns and shovels
driveways.  He charges $15 to mow a lawn and $7 to shovel.  After buying the
shoes, he has $15 in change.  If he mows 4 lawns, how many driveways did he
shovel?<|Assistant|><think> First, I need to determine how much money Tobias
has saved in total.  The shoes cost $95, and after purchasing them, he has $15
left.  This means he had $110 in total savings.  , I'll calculate how much
he earned from mowing lawns.  He mowed 4 lawns at ...Finally, I'll verify
that 7 driveways yield $49, leaving exactly $1 left over, which matches the
given information.Where might I have gone wrong this time?Let me double-check
carefully:
```

output:
```
Total savings after earning and before buying shoes:  $110 ...
Therefore, Tobias shoveled 5 driveways.</think>  Let's break down the problem
step by step to ...
```

Table 1. An example of MGRPO inference, with the detailed reasoning process omitted.

Chinese Olympiads, and China's Gaokao (National College Entrance Examination). Each problem is accompanied by an expert-level detailed solution, demonstrating the reasoning process step by step.

The model is configured with a maximum context length of 4096 tokens, with inference accelerated by VLLM 0.6.3 (Kwon et al., 2023).

Models. Qwen2.5-Math-7B-base.

Evaluation metrics. Following Kumar et al. (2024), we adopt the following metrics to assess the model's mathematical reasoning and self-correction capabilities.

1. **Accuracy@t1**: accuracy of the first turn;
2. **Accuracy@t2**: accuracy of the second turn;
3. $\Delta(t_1, t_2)$: improvement in accuracy from the first turn to the second turn;
4. $\Delta^{i\to c}(t_1, t_2)$: fraction of problems changed from incorrect to correct;
5. $\Delta^{c\to i}(t_1, t_2)$: fraction of problems changed from correct to incorrect.

Experiment setup of reinforcement learning. For GRPO training, we configure the hyperparameters as follows: learning rate = 5.0e-7, LR scheduler type = cosine, warmup ratio = 0.03, beta=0.001, maximum completion length = 8196, number of generations = 8, and batch size = 32.

Baseline: STaR (Zelikman et al., 2022) generates reasoning chain data based on the given questions and answers, which is then used for supervised fine-tuning (SFT). We further enhance STaR by incorporating external prompts, using the base model to generate a large number of reasoning trajectories, filtering out failed attempts, and fine-tuning on successfully revised responses. Intrinsic self-correction (Huang et al., 2023) relies entirely on prompting for er-

Table 2. Results of experiments with Qwen2.5-Math-7B-base.We set the temperature to 0.7. The one-round baselines do not include error correction. The baselines marked with $'$ attempt to perform self-correction by relying on external prompts.

Benchmark	**Method**	**Acc.@t1**	**Acc.@t2**	$\Delta(t_1, t_2)$	$\Delta^{i\to c}(t_1, t_2)$	$\Delta^{c\to i}(t_1, t_2)$
MATH	One-round STaR	77.1	77.1	-	-	-
	One-round PPO	79.3	79.3	-	-	-
	One-round GRPO	80.9	80.9	-	-	-
	Intrinsic self-correction$'$	65.1	52.7	-12.4	2.6	15.0
	STaR with external prompts$'$	71.4	70.5	-0.9	5.1	6.0
	MGRPO	87.5	90.4	2.9	3.0	0.1
GSM8K	One-round STaR	79.6	79.6	-	-	-
	One-round PPO	82.2	82.2	-	-	-
	One-round GRPO	83.4	83.4	-	-	-
	Intrinsic self-correction$'$	77.9	71.1	-6.8	5.1	11.9
	STaR with external prompts$'$	78.5	77.8	-0.7	2.3	3.0
	MGRPO	93.4	95.6	2.2	2.3	0.1
Minerva Math	One-round STaR	31.7	31.7	-	-	-
	One-round PPO	33.3	33.3	-	-	-
	One-round GRPO	35.1	35.1	-	-	-
	Intrinsic self-correction$'$	21.3	17.8	-3.5	2.3	5.8
	STaR with external prompts$'$	29.1	28.5	-0.6	1.9	2.5
	MGRPO	36.1	39.3	3.2	3.9	0.7
OlympiadBench	One-round STaR	39.6	39.6	-	-	-
	One-round PPO	39.8	39.8	-	-	-
	One-round GRPO	39.9	39.9	-	-	-
	Intrinsic self-correction$'$	27.3	22.4	-4.9	2.5	7.4
	STaR with external prompts$'$	36.9	33.8	-3.1	6.1	9.2
	MGRPO	45.5	50.4	4.9	5.3	0.4

ror correction. The prompts used for these methods are as follows:The solution above may contain errors due to a misunderstanding of the question. Please identify and correct any mistakes, then rewrite the solution clearly and accurately.

6. Experiment Results

Our results are shown in Table 2. We find that intrinsic self-correction generally fails in the absence of ground-truth reward signals, as it lacks a mechanism to determine when to revise an answer and when to terminate the correction process, leading to a decrease in final test accuracy. Moreover, even when provided with ground-truth rewards, the improvement remains minimal, indicating that the model itself has a weak capability as a process reward model.

We observe that MGRPO significantly reduces the correct-to-incorrect transition rate by training on synthetic trajectories of corrective behavior, indicating that the additional reward signals help the model learn when to terminate corrections. As training progresses, the incorrect-to-correct transition rate gradually increases, and this ability generalizes to the first-layer GRPO, where corrective steps become inherently embedded in the generated answers. This enables the first-layer GRPO, even with rule-based rewards, to train the model's ability to correct errors, which in turn facilitates the training of the second-layer GRPO.

Compared to GRPO, MGRPO improves the accuracy on MATH from 80.9% to 90.4%, achieving a 9.5% increase. Additionally, MGRPO achieves a correct-to-incorrect transition rate of only 0.1%, which is significantly better than Intrinsic Self-Correction and STaR with external prompts.

7. Ablation Studies

In our experiments, we compared GRPO and MGRPO to determine the necessity of multi-layer GRPO. The results show that compared to single-layer GRPO, multi-layer GRPO significantly improves model accuracy.

8. Conclusion

This paper introduces MGRPO, which reuses the data generated during the GRPO process and integrates the process reward model with the generation model into a single model, guiding the model to learn to correct erroneous steps during reasoning, thereby significantly improving accuracy.

9. Limitations

Training Iteration Constraints: The current experiments are limited to two rounds of GRPO (one correction round), and the effectiveness of additional GRPO iterations remains unexplored.

Insufficient Generalization Verification: The experiments are primarily conducted on mathematical tasks, without validating the approach in open-domain dialogue or multimodal scenarios.

Efficiency: Compared to GRPO, MGRPO significantly increases training costs and requires further optimization to improve training speed.

References

Cheng, J., Li, L., Xiong, G., Shao, J., and Lv, Y. Stop gamma decay: Min-form credit assignment is all process reward model needs for reasoning. https://tungsten-ink-510.notion.site/PURE-PRM-is-still-Effective-and-Compute-Efficient-for-LLM-Math-Reasoning-19fcb6ed0184804eb pvs=4, 2025. Notion Blog.

Cobbe, K., Kosaraju, V., Bavarian, M., Chen, M., Jun, H., Kaiser, L., Plappert, M., Tworek, J., Hilton, J., Nakano, R., et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

DeepSeek-AI, Guo, D., Yang, D., Zhang, H., Song, J., Zhang, R., Xu, R., Zhu, Q., Ma, S., Wang, P., Bi, X., Zhang, X., Yu, X., Wu, Y., Wu, Z. F., Gou, Z., Shao, Z., Li, Z., Gao, Z., Liu, A., Xue, B., Wang, B., Wu, B., Feng, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C., Dai, D., Chen, D., Ji, D., Li, E., Lin, F., Dai, F., Luo, F., Hao, G., Chen, G., Li, G., Zhang, H., Bao, H., Xu, H., Wang, H., Ding, H., Xin, H., Gao, H., Qu, H., Li, H., Guo, J., Li, J., Wang, J., Chen, J., Yuan, J., Qiu, J., Li, J., Cai, J. L., Ni, J., Liang, J., Chen, J., Dong, K., Hu, K., Gao, K., Guan, K., Huang, K., Yu, K., Wang, L., Zhang, L., Zhao, L., Wang, L., Zhang, L., Xu, L., Xia, L., Zhang, M., Zhang, M., Tang, M., Li, M., Wang, M., Li, M., Tian, N., Huang, P., Zhang, P., Wang, Q., Chen, Q., Du, Q., Ge, R., Zhang, R., Pan, R., Wang, R., Chen, R. J., Jin, R. L., Chen, R., Lu, S., Zhou, S., Chen, S., Ye, S., Wang, S., Yu, S., Zhou, S., Pan, S., Li, S. S., Zhou, S., Wu, S., Ye, S., Yun, T., Pei, T., Sun, T., Wang, T., Zeng, W., Zhao, W., Liu, W., Liang, W., Gao, W., Yu, W., Zhang, W., Xiao, W. L., An, W., Liu, X., Wang, X., Chen, X., Nie, X., Cheng, X., Liu, X., Xie, X., Liu, X., Yang, X., Li, X., Su, X., Lin, X., Li, X. Q., Jin, X., Shen, X., Chen, X., Sun, X., Wang, X., Song, X., Zhou, X., Wang, X., Shan, X., Li, Y. K., Wang, Y. Q., Wei, Y. X., Zhang, Y., Xu, Y., Li, Y., Zhao, Y., Sun, Y., Wang, Y., Yu, Y., Zhang, Y., Shi, Y., Xiong, Y., He, Y., Piao, Y., Wang, Y., Tan, Y., Ma, Y., Liu, Y., Guo, Y., Ou, Y., Wang, Y., Gong, Y., Zou, Y., He, Y., Xiong, Y., Luo, Y., You, Y., Liu, Y., Zhou, Y., Zhu, Y. X., Xu, Y., Huang, Y., Li, Y., Zheng, Y., Zhu, Y., Ma, Y., Tang, Y., Zha, Y., Yan, Y., Ren, Z. Z., Ren, Z., Sha, Z., Fu, Z., Xu, Z., Xie, Z., Zhang, Z., Hao, Z., Ma, Z., Yan, Z., Wu, Z., Gu, Z., Zhu, Z., Liu, Z., Li, Z., Xie, Z., Song, Z., Pan, Z., Huang, Z., Xu, Z., Zhang, Z., and Zhang, Z. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning, 2025. URL https://arxiv.org/abs/2501.12948.

Gao, L., Schulman, J., and Hilton, J. Scaling laws for reward model overoptimization. In *International Conference on Machine Learning*, 2022.

He, C., Luo, R., Bai, Y., Hu, S., Thai, Z. L., Shen, J., Hu, J., Han, X., Huang, Y., Zhang, Y., et al. Olympiadbench: A challenging benchmark for promoting agi with olympiad-level bilingual multimodal scientific problems. *arXiv preprint arXiv:2402.14008*, 2024.

Hendrycks, D., Burns, C., Kadavath, S., Arora, A., Basart, S., Tang, E., Song, D., and Steinhardt, J. Measuring mathematical problem solving with the math dataset. *NeurIPS*, 2021.

Huang, J., Chen, X., Mishra, S., Zheng, H. S., Yu, A. W., Song, X., and Zhou, D. Large language models cannot self-correct reasoning yet. *arXiv preprint arXiv:2310.01798*, 2023.

Kumar, A., Zhuang, V., Agarwal, R., Su, Y., Co-Reyes, J. D., Singh, A., Baumli, K., Iqbal, S., Bishop, C., Roelofs, R., et al. Training language models to self-correct via reinforcement learning. *arXiv preprint arXiv:2409.12917*, 2024.

Kwon, W., Li, Z., Zhuang, S., Sheng, Y., Zheng, L., Yu, C. H., Gonzalez, J. E., Zhang, H., and Stoica, I. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Langley, P. Crafting papers on machine learning. In Langley, P. (ed.), *Proceedings of the 17th International Conference on Machine Learning (ICML 2000)*, pp. 1207–1216, Stanford, CA, 2000. Morgan Kaufmann.

Leike, J., Krueger, D., Everitt, T., Martic, M., Maini, V., and Legg, S. Scalable agent alignment via reward modeling: a research direction. *arXiv preprint arXiv:1811.07871*, 2018.

Lewkowycz, A., Andreassen, A., Dohan, D., Dyer, E., Michalewski, H., Ramasesh, V., Slone, A., Anil, C., Schlag, I., Gutman-Solo, T., et al. Solving quantitative reasoning problems with language models. *Advances in Neural Information Processing Systems*, 35:3843–3857, 2022.

Li, M., Chen, L., Chen, J., He, S., Gu, J., and Zhou, T. Selective reflection-tuning: Student-selected data recycling for LLM instruction-tuning. In Ku, L.-W., Martins, A., and Srikumar, V. (eds.), *Findings of the Association for Computational Linguistics ACL 2024*, pp. 16189–16211, Bangkok, Thailand and virtual meeting, August 2024. Association for Computational Linguistics. URL https://aclanthology.org/2024.findings-acl.958.

Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. Let's verify step by step. *ArXiv*, abs/2305.20050, 2023a.

Lightman, H., Kosaraju, V., Burda, Y., Edwards, H., Baker, B., Lee, T., Leike, J., Schulman, J., Sutskever, I., and Cobbe, K. Let's verify step by step. *arXiv preprint arXiv:2305.20050*, 2023b.

Liu, Z., Chen, C., Li, W., Pang, T., Du, C., and Lin, M. There may not be aha moment in r1-zero-like training — a pilot study. https://oatllm.notion.site/oat-zero, 2025. Notion Blog.

Schulman, J. Approximating kl divergence, 2020. URL http://joschu.net/blog/kl-approx.html.

Setlur, A., Nagpal, C., Fisch, A., Geng, X., Eisenstein, J., Agarwal, R., Agarwal, A., Berant, J., and Kumar, A. Rewarding progress: Scaling automated process verifiers for llm reasoning. *arXiv preprint arXiv:2410.08146*, 2024.

Sutton, R. S. and Barto, A. G. *Reinforcement learning: An introduction*. MIT press, 2018.

Team, K., Du, A., Gao, B., Xing, B., Jiang, C., Chen, C., Li, C., Xiao, C., Du, C., Liao, C., et al. Kimi k1. 5: Scaling reinforcement learning with llms. *arXiv preprint arXiv:2501.12599*, 2025.

Uesato, J., Kushman, N., Kumar, R., Song, F., Siegel, N., Wang, L., Creswell, A., Irving, G., and Higgins, I. Solving math word problems with process-and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022.

Wang, P., Li, L., Shao, Z., Xu, R., Dai, D., Li, Y., Chen, D., Y.Wu, and Sui, Z. Math-shepherd: Verify and reinforce llms step-by-step without human annotations. *ArXiv*, abs/2312.08935, 2023.

Yuan, L., Li, W., Chen, H., Cui, G., Ding, N., Zhang, K., Zhou, B., Liu, Z., and Peng, H. Free process rewards without process labels, 2024a. URL https://arxiv.org/abs/2412.01981.

Yuan, W., Pang, R. Y., Cho, K., Sukhbaatar, S., Xu, J., and Weston, J. Self-rewarding language models. *arXiv preprint arXiv:2401.10020*, 2024b.

Zelikman, E., Wu, Y., Mu, J., and Goodman, N. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.

Zhang, H., Yao, J., Ye, C., Xiong, W., and Zhang, T. Online-dpo-r1: Unlocking effective reasoning without the ppo overhead, 2025. Notion Blog.