Methods	Data Source	Reward Function	Gradient Coefficient
SFT	$q, o \sim P_{sft}(Q, O)$	-	1
RFT	$q \sim P_{sft}(Q),\ o \sim \pi_{sft}(O\|q)$	Rule	Equation 10
DPO	$q \sim P_{sft}(Q),\ o^+, o^- \sim \pi_{sft}(O\|q)$	Rule	Equation 14
Online RFT	$q \sim P_{sft}(Q),\ o \sim \pi_\theta(O\|q)$	Rule	Equation 10
PPO	$q \sim P_{sft}(Q),\ o \sim \pi_\theta(O\|q)$	Model	Equation 18
GRPO	$q \sim P_{sft}(Q),\ \{o_i\}_{i=1}^G \sim \pi_\theta(O\|q)$	Model	Equation 21

Table 10 | The data source and gradient coefficient of different methods. P_{sft} denotes the data distribution of supervised fine-tuning datasets. $\pi_{\theta_{sft}}$ and π_θ denote the supervised fine-tuned model and the real-time policy model during the online training process, respectively.

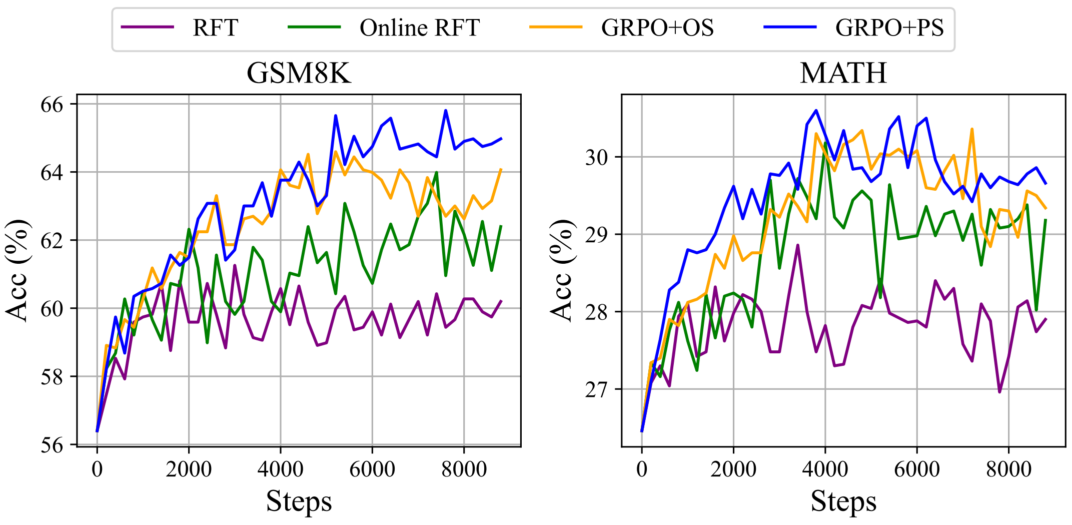

Figure 5 | Performance of the DeepSeekMath-Instruct 1.3B model, which was further trained using various methods, on two benchmarks.

- **Rejection Sampling Fine-tuning (RFT)**: RFT further fine-tunes the SFT model on the filtered outputs sampled from the SFT model based on SFT questions. RFT filters the outputs based on the correctness of their answers.
- **Direct Preference Optimization (DPO)**: DPO further refines the SFT model by fine-tuning it on augmented outputs sampled from the SFT model, using pair-wise DPO loss.
- **Online Rejection Sampling Fine-tuning (Online RFT)**: Different from RFT, Online RFT initiates the policy model using the SFT model and refines it by fine-tuning with the augmented outputs sampled from the real-time policy model.
- **PPO/GRPO**: PPO/GRPO initializes the policy model using the SFT model and reinforces it with the outputs sampled from the real-time policy model.

We summarize the components of these methods in Table 10. Please refer to Appendix A.1 for a more detailed derivation process.

Observation about Data Source We divide the data source into two categories, online sampling, and offline sampling. Online sampling denotes that the training data is from the exploration results of the real-time training policy model, while offline sampling denotes that the

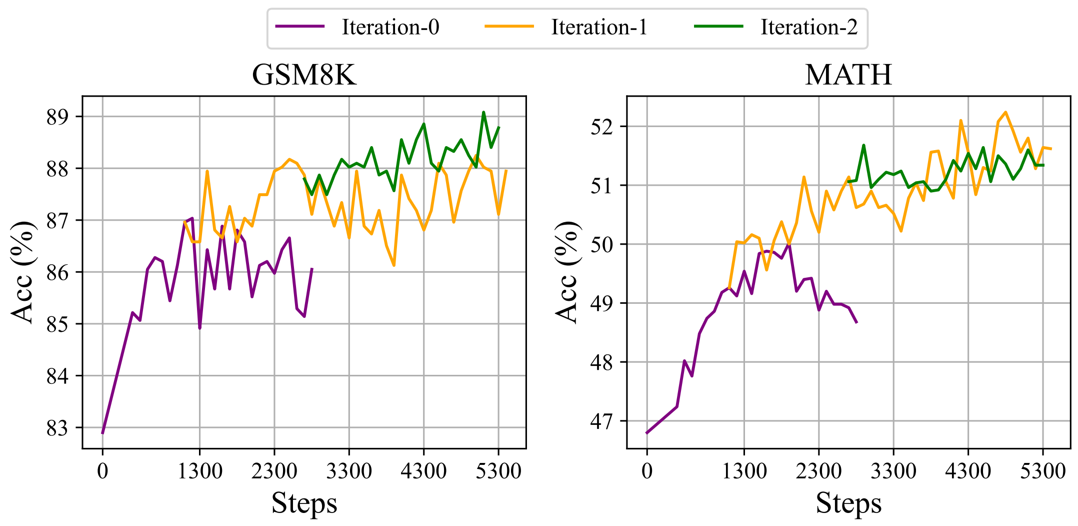

Figure 6 | Performance of iterative reinforcement learning with DeepSeekMath-Instruct 7B on two benchmarks.

training data is from the sampling results of the initial SFT model. RFT and DPO follow the offline style, while Online RFT and GRPO follow the online style.

As shown in Figure 5, we find that the Online RFT significantly outperforms RFT on two benchmarks. Specifically, Online RFT is comparable to RFT in the early stage of training but gains an absolute advantage in the later stage, demonstrating the superiority of online training. This is intuitive, as in the initial stage, the actor and the SFT model exhibit close resemblance, with the sampled data revealing only minor differences. In the later stage, however, the data sampled from the actor will exhibit more significant differences, and real-time data sampling will offer greater advantages.

Observation about Gradient Coefficient The algorithm processes the reward signal to the gradient coefficient to update the model parameter. We divide the reward function as 'Rule' and 'Model' in our experiments. Rule refers to judging the quality of a response based on the correctness of the answer, and Model denotes that we train a reward model to score each response. The training data of the reward model is based on the rule judgment. Equations 10 and 21 highlight a key difference between GRPO and Online RFT: GRPO uniquely adjusts its gradient coefficient based on the reward value provided by the reward model. This allows for differential reinforcement and penalization of responses according to their varying magnitudes. In contrast, Online RFT lacks this feature; it does not penalize incorrect responses and uniformly reinforces all responses with correct answers at the same level of intensity.

As demonstrated in Figure 5, GRPO surpasses online RFT, thereby highlighting the efficiency of altering positive and negative gradient coefficients. In addition, GRPO+PS shows superior performance compared to GRPO+OS, indicating the benefits of using fine-grained, step-aware gradient coefficients. Furthermore, we explore the iterative RL, in our experiments, we conduct two rounds of iteration. As shown in Figure 6, we notice that the iterative RL significantly improves the performance, especially at the first iteration.

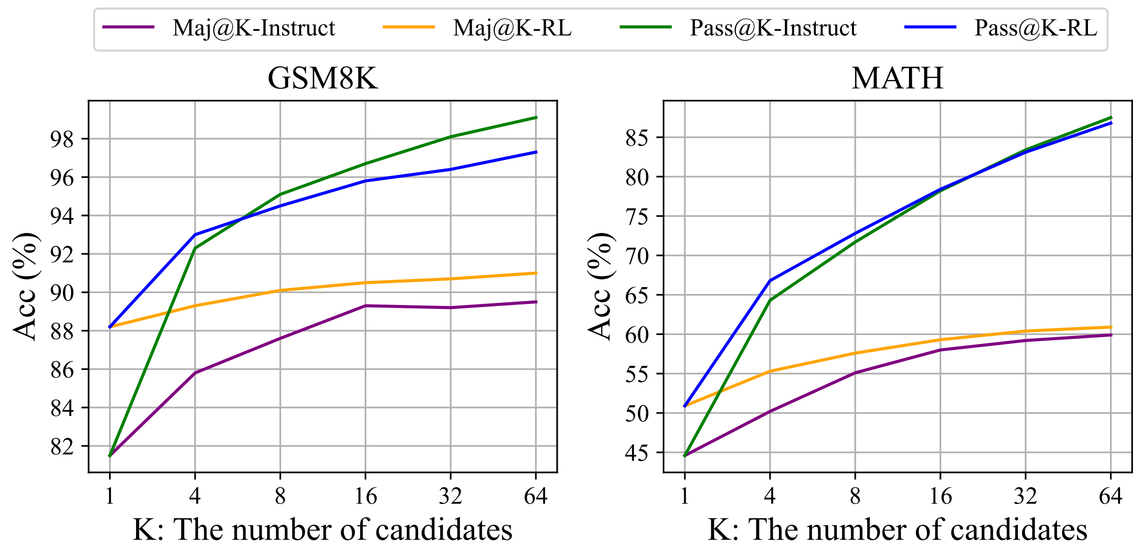

Figure 7 | The Maj@K and Pass@K of SFT and RL DeepSeekMath 7B on GSM8K and MATH (temperature 0.7). It was noted that RL enhances Maj@K but not Pass@K.

5.2.2. *Why RL Works?*

In this paper, we conduct reinforcement learning based on a subset of instruction tuning data, and it achieves significant performance enhancement upon the instruction tuning model. To further explain why reinforcement learning works. We evaluate the Pass@K and Maj@K accuracy of the Instruct and RL models on two benchmarks. As shown in Figure 7, RL enhances Maj@K's performance but not Pass@K. These findings indicate that RL enhances the model's overall performance by rendering the output distribution more robust, in other words, **it seems that the improvement is attributed to boosting the correct response from TopK rather than the enhancement of fundamental capabilities.** Similarly, (Wang et al., 2023a) identified a **misalignment problem** in reasoning tasks within the SFT model, showing that the reasoning performance of SFT models can be improved through a series of preference alignment strategies (Song et al., 2023; Wang et al., 2023a; Yuan et al., 2023b).

5.2.3. *How to Achieve More Effective RL?*

We demonstrate RL works pretty well in mathematical reasoning tasks. We also provide a unified paradigm to understand different representative training methods. Within this paradigm, all methods are conceptualized as either direct or simplified RL techniques. As summarized in Equation 5, there exist three key components: Data Source, Algorithm, and Reward Function. We provide some potential future directions about the three components.

Data Source Data source is the raw material of all training methods. In the context of RL, we specifically refer to the data source as the unlabeled questions with the outputs sampled from the policy model. In this paper, we only use the questions from the instruction tuning stage and a naive nucleus sampling to sample outputs. We think this is a potential reason that our RL pipeline only improves the Maj@K performance. In the future, we will explore our RL pipeline on out-of-distribution question prompts, in conjunction with **advanced sampling (decoding) strategies**, like those based on tree-search methods (Yao et al., 2023). Also, the **efficient inference techniques** (Kwon et al., 2023; Leviathan et al., 2023; Xia et al., 2023, 2024), which determines