DeepSeekMath

Soumyabrata Chaudhuri (Soumya)

TEXAS
The University of Texas at Austin

Contributions

- Introduces **DeepSeekMath-7B**, a SOTA math reasoning model.
- A meticulous **Iterative data mining pipeline** for crawling the web.
- Introduces **Group Relative Policy Optimization** (GRPO).

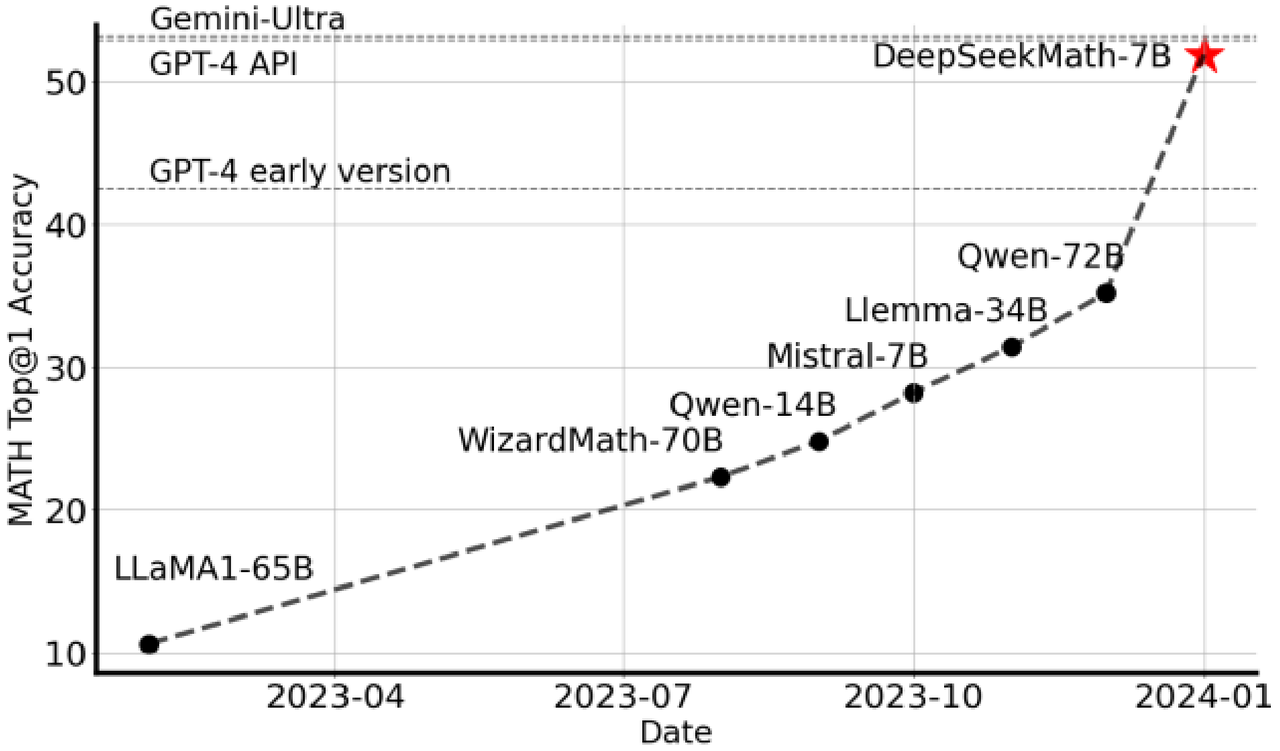


Figure 1 | Top1 accuracy of open-source models on the competition-level MATH benchmark (Hendrycks et al., 2021) without the use of external toolkits and voting techniques.

Pretraining Data & Corpus Construction

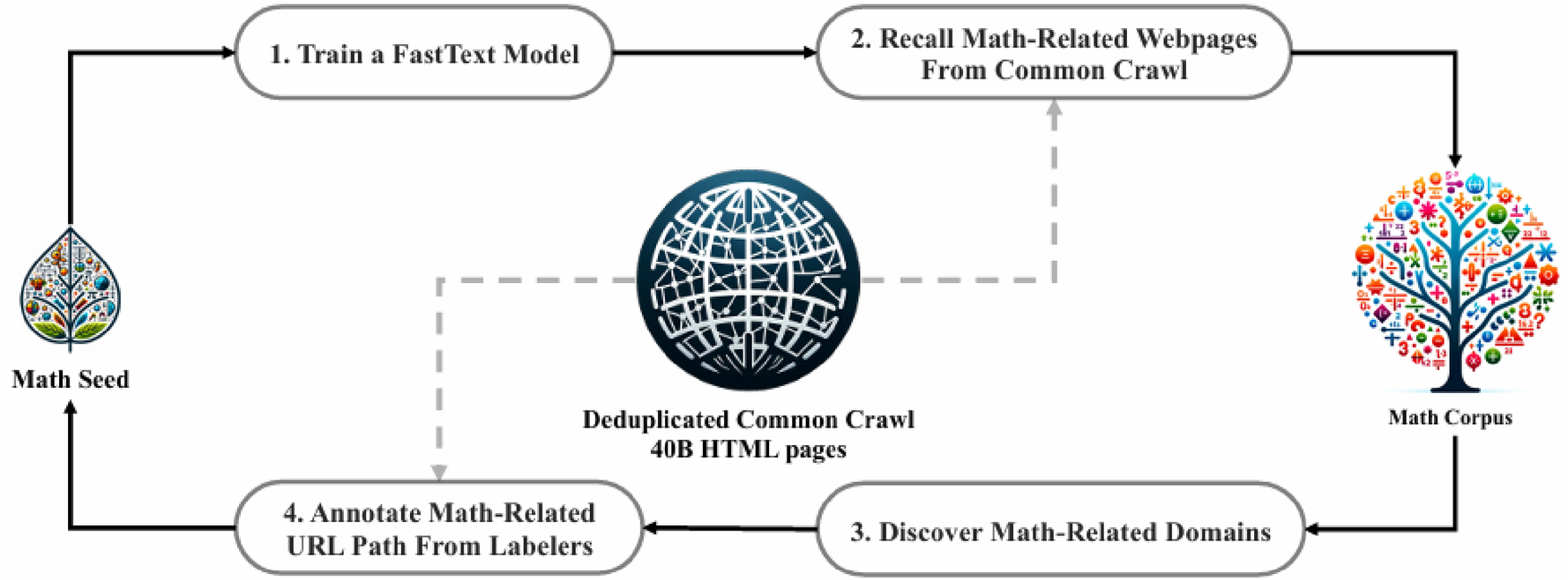


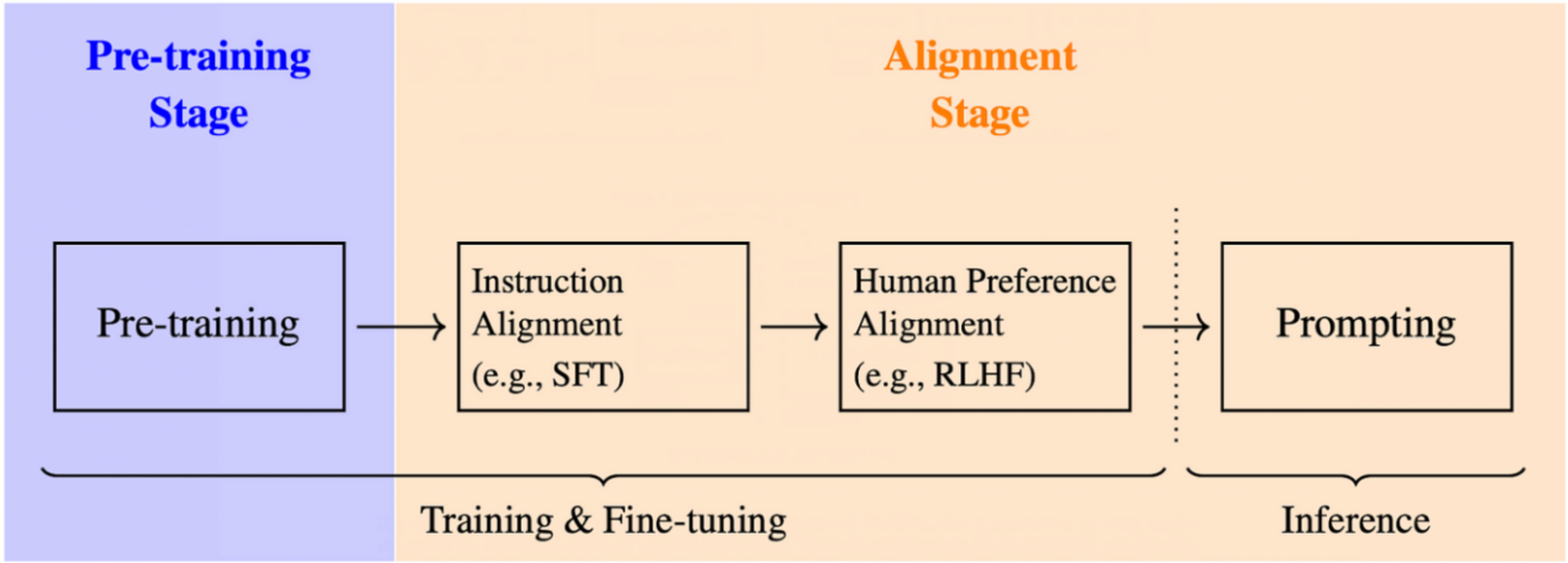
Figure 2 | An iterative pipeline that collects mathematical web pages from Common Crawl.

Pretraining Data & Corpus Construction

Math Corpus	Size	English Benchmarks					Chinese Benchmarks		
		GSM8K	MATH	OCW	SAT	MMLU STEM	CMATH	Gaokao MathCloze	Gaokao MathQA
No Math Training	N/A	2.9%	3.0%	2.9%	15.6%	19.5%	12.3%	0.8%	17.9%
MathPile	8.9B	2.7%	3.3%	2.2%	12.5%	15.7%	1.2%	0.0%	2.8%
OpenWebMath	13.6B	11.5%	8.9%	3.7%	31.3%	29.6%	16.8%	0.0%	14.2%
Proof-Pile-2	51.9B	14.3%	11.2%	3.7%	43.8%	29.2%	19.9%	5.1%	11.7%
DeepSeekMath Corpus	**120.2B**	**23.8%**	**13.6%**	**4.8%**	**56.3%**	**33.1%**	**41.5%**	**5.9%**	**23.6%**

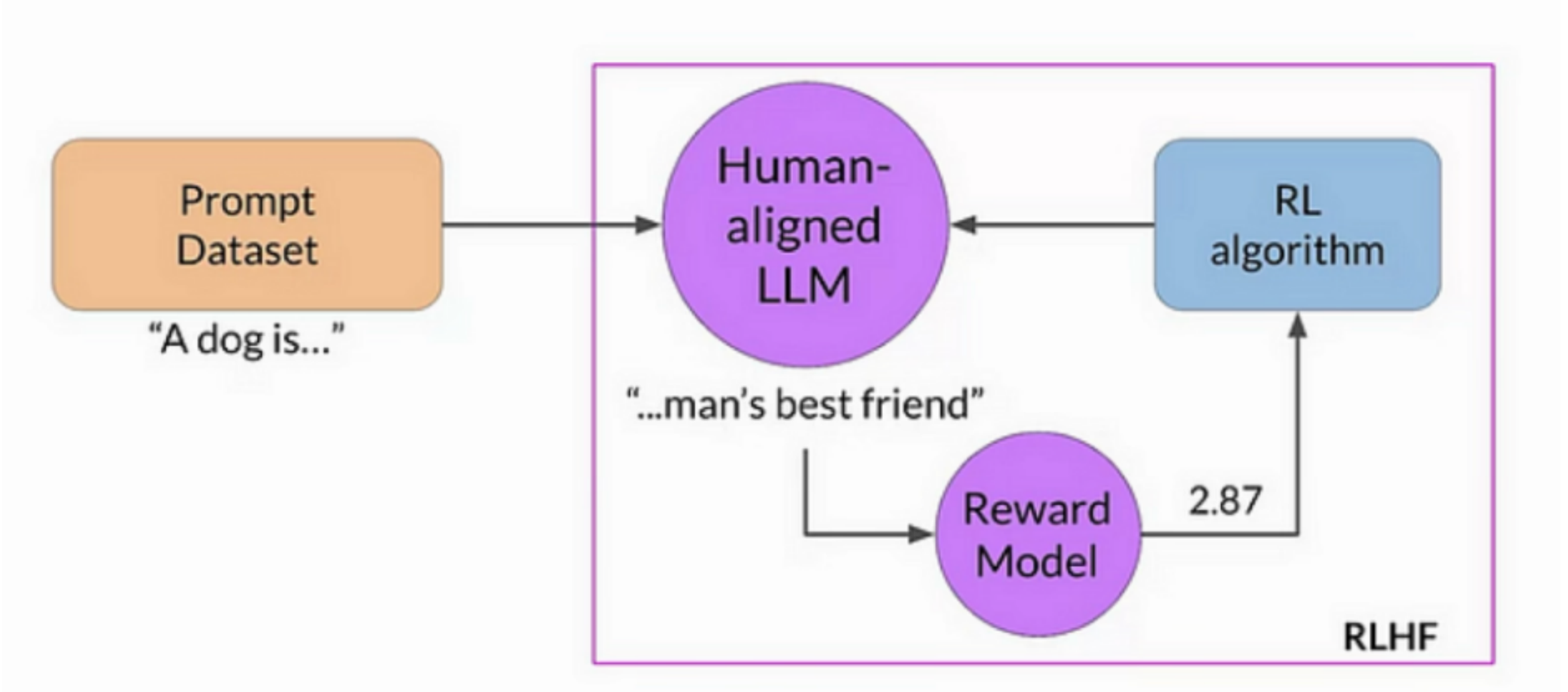
Table 1 | Performance of DeepSeek-LLM 1.3B trained on different mathematical corpora, evaluated using few-shot chain-of-thought prompting. Corpus sizes are calculated using our tokenizer with a vocabulary size of 100K.

Three stages of LLM training



The 3 steps of LLM training [1]

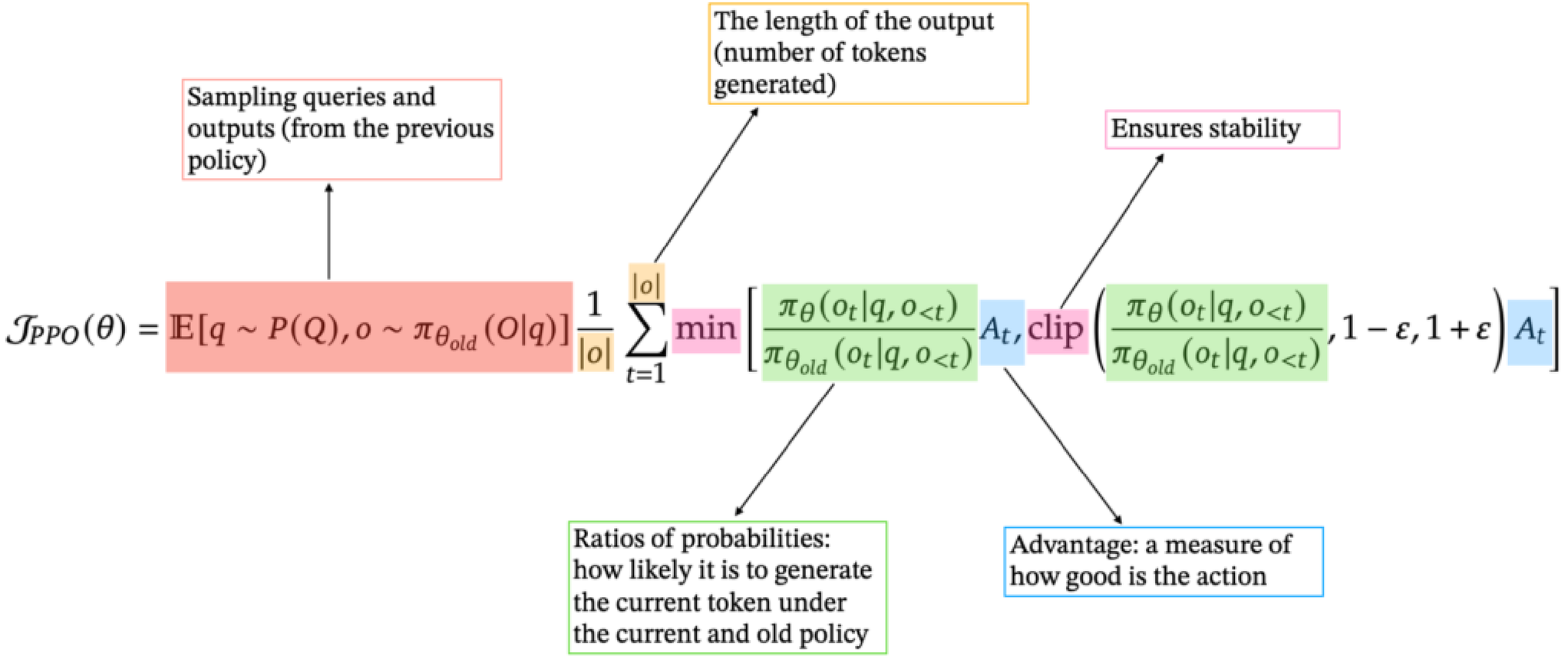
RL in the context of LLMs



Simplified RLHF Process [3]

[4]

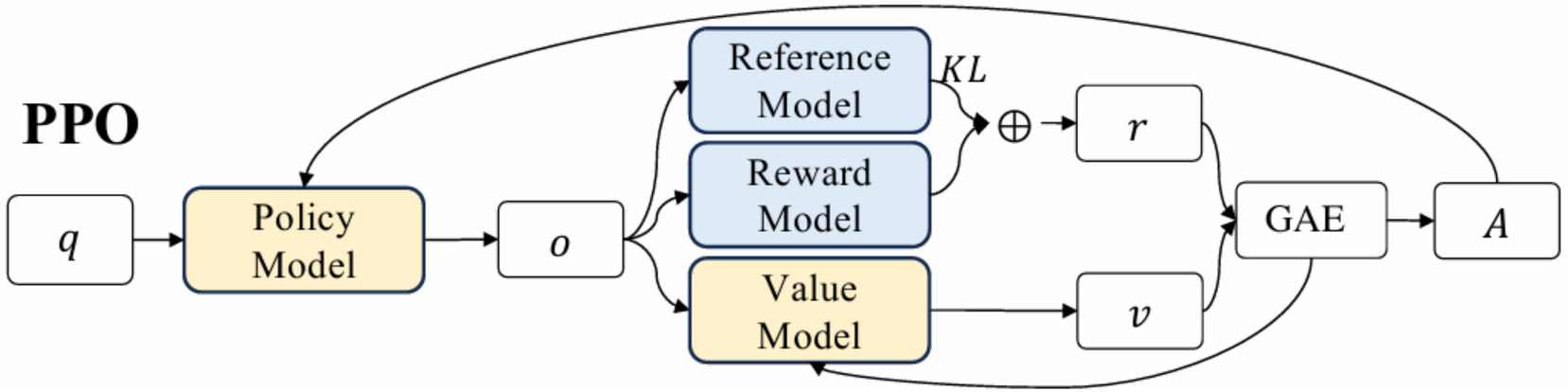
Proximal Policy Optimization (PPO)

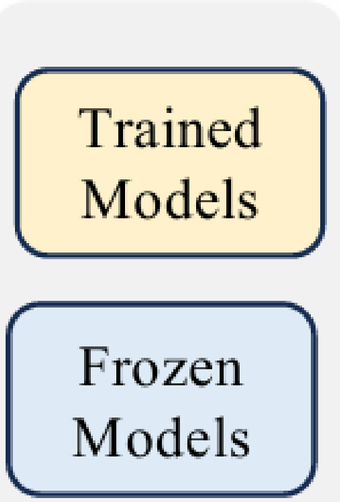


$$r_t = r_\varphi(q, o_{\leq t}) - \beta \log \frac{\pi_\theta(o_t|q,o_{<t})}{\pi_{ref}(o_t|q,o_{<t})}$$

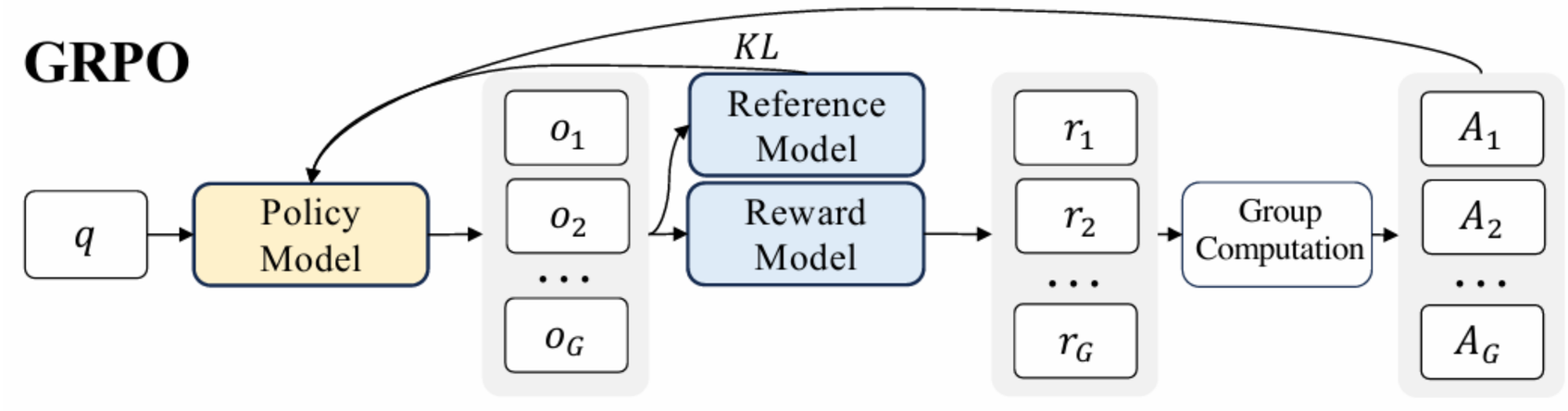
[4]

PPO to GRPO


PPO
q
Policy Model
o
Reference Model
Reward Model
Value Model
KL
⊕
r
v
GAE
A


Trained Models
Frozen Models

[4]


GRPO
KL
q
Policy Model
o_1
o_2
o_G
Reference Model
Reward Model
r_1
r_2
r_G
Group Computation
A_1
A_2
A_G

Group Relative Policy Optimization (GRPO)

Sample queries as before but now, sample a group of G outputs for each query

New KL divergence term to ensure stability relative to the reference policy (i.e. model before the RL process)

$$\mathcal{J}_{GRPO}(\theta) = \mathbb{E}[q \sim P(Q), \{o_i\}_{i=1}^{G} \sim \pi_{\theta_{old}}(O|q)]$$

$$\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\left\{\min\left[\frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q,o_{i,<t})}\hat{A}_{i,t}, \text{clip}\left(\frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{old}}(o_{i,t}|q,o_{i,<t})}, 1-\varepsilon, 1+\varepsilon\right)\hat{A}_{i,t}\right] - \beta\mathbb{D}_{KL}\left[\pi_\theta||\pi_{ref}\right]\right\}$$

This is the length of the i^{th} output of the group

Exactly same logic as PPO, with the new advantage function 🙂

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

[4]

GRPO Variants

- Outcome Supervision RL:

$$\hat{A}_{i,t} = \tilde{r}_i = \frac{r_i - \text{mean}(\mathbf{r})}{\text{std}(\mathbf{r})}$$

[4]

- Process Supervision RL:

$$\mathbf{R} = \{\{r_1^{index(1)}, \cdots, r_1^{index(K_1)}\}, \cdots, \{r_G^{index(1)}, \cdots, r_G^{index(K_G)}\}\}$$

$$\tilde{r}_i^{index(j)} = \frac{r_i^{index(j)} - \text{mean}(\mathbf{R})}{\text{std}(\mathbf{R})}$$

$$\hat{A}_{i,t} = \sum_{index(j) \geq t} \tilde{r}_i^{index(j)}$$

GRPO Variants

Algorithm 1 Iterative Group Relative Policy Optimization

Input initial policy model $\pi_{\theta_{\text{init}}}$; reward models r_φ; task prompts $\mathcal{D}$; hyperparameters ε, β, μ

policy model $\pi_\theta \leftarrow \pi_{\theta_{\text{init}}}$
for iteration = 1, . . . , I **do**
    reference model $\pi_{ref} \leftarrow \pi_\theta$
    for step = 1, . . . , M **do**
        Sample a batch $\mathcal{D}_b$ from $\mathcal{D}$
        Update the old policy model $\pi_{\theta_{old}} \leftarrow \pi_\theta$
        Sample G outputs $\{o_i\}_{i=1}^{G} \sim \pi_{\theta_{old}}(\cdot \mid q)$ for each question $q \in \mathcal{D}_b$
        Compute rewards $\{r_i\}_{i=1}^{G}$ for each sampled output o_i by running r_φ
        Compute $\hat{A}_{i,t}$ for the t-th token of o_i through group relative advantage estimation.
        for GRPO iteration = 1, . . . , μ **do**
            Update the policy model π_θ by maximizing the GRPO objective (Equation 21)
    Update r_φ through continuous training using a replay mechanism.

Output π_θ

[4]

Discussion: Code Training Benefits Mathematical Reasoning

Training Setting	Training Tokens			w/o Tool Use			w/ Tool Use	
	General	Code	Math	GSM8K	MATH	CMATH	GSM8K+Python	MATH+Python
No Continual Training	–	–	–	2.9%	3.0%	12.3%	2.7%	2.3%
Two-Stage Training								
Stage 1: General Training	400B	–	–	2.9%	3.2%	14.8%	3.3%	2.3%
Stage 2: Math Training	–	–	150B	19.1%	14.4%	37.2%	14.3%	6.7%
Stage 1: Code Training	–	400B	–	5.9%	3.6%	19.9%	12.4%	10.0%
Stage 2: Math Training	–	–	150B	**21.9%**	**15.3%**	**39.7%**	17.4%	9.4%
One-Stage Training								
Math Training	–	–	150B	20.5%	13.1%	37.6%	11.4%	6.5%
Code & Math Mixed Training	–	400B	150B	17.6%	12.1%	36.3%	**19.7%**	**13.5%**

Table 6 | Investigation of how code affects mathematical reasoning under different training settings. We experiment with DeepSeek-LLM 1.3B, and evaluate its mathematical reasoning performance without and with tool use via few-shot chain-of-thought prompting and few-shot program-of-thought prompting, respectively.

[4]

Discussion: Arxiv Papers seem Ineffective in improving Math Reasoning

Model	Size	ArXiv Corpus	English Benchmarks					Chinese Benchmarks		
			GSM8K	MATH	OCW	SAT	MMLU STEM	CMATH	Gaokao MathCloze	Gaokao MathQA
DeepSeek-LLM	1.3B	No Math Training	2.9%	3.0%	2.9%	15.6%	19.5%	12.3%	0.8%	17.9%
		MathPile	2.7%	3.3%	2.2%	12.5%	15.7%	1.2%	0.0%	2.8%
		ArXiv-RedPajama	3.3%	3.4%	4.0%	9.4%	9.0%	7.4%	0.8%	2.3%
DeepSeek-Coder-Base-v1.5	7B	No Math Training	29.0%	12.5%	6.6%	40.6%	38.1%	45.9%	5.9%	21.1%
		MathPile	23.6%	11.5%	7.0%	46.9%	35.8%	37.9%	4.2%	25.6%
		ArXiv-RedPajama	28.1%	11.1%	7.7%	50.0%	35.2%	42.6%	7.6%	24.8%

Table 8 | Effect of math training on different arXiv datasets. Model performance is evaluated with few-shot chain-of-thought prompting.

[4]

ArXiv Corpus	miniF2F-valid	miniF2F-test
No Math Training	20.1%	21.7%
MathPile	16.8%	16.4%
ArXiv-RedPajama	14.8%	11.9%

Table 9 | Effect of math training on different arXiv corpora, the base model being DeepSeek-Coder-Base-v1.5 7B. We evaluate informal-to-formal proving in Isabelle.

Discussion: Why RL works?

- RL enhances Maj@K's performance but not Pass@K.
- It seems that the improvement is attributed to boosting the correct response from TopK rather than the enhancement of fundamental capabilities.

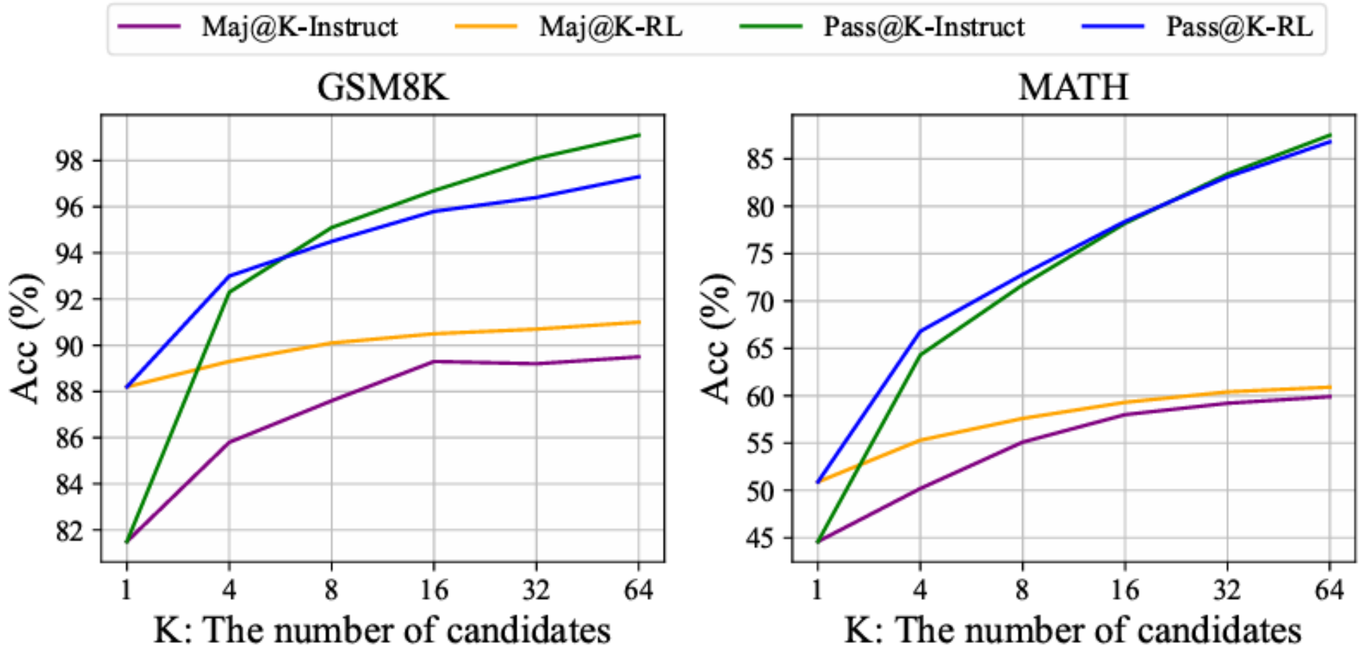


Figure 7 | The Maj@K and Pass@K of SFT and RL DeepSeekMath 7B on GSM8K and MATH (temperature 0.7). It was noted that RL enhances Maj@K but not Pass@K.

[4]

Conclusion

- **Data Curation:** DeepSeekMath surpasses all open-source models on the MATH benchmark and nears closed-source model performance through large-scale training with rich mathematical web data.

[4]

- **Algorithm:** They introduce **Group Relative Policy Optimization (GRPO)** which effectively polishes its response distribution with lower memory usage compared to PPO.

- **Limitations & Future Work:** The model struggles with geometry and theorem-proof tasks and lacks strong few-shot learning ability. Their future work aims to refine data selection and reinforcement learning approaches to address these gaps.


THANK YOU!