



Group Relative Policy Optimization

▼ Introduction



전체적인 흐름

DeepSeekMath 논문에서 제시한 RL 알고리즘 GRPO를 중심으로 리뷰

GRPO 자체는 LLM RL에 초점이 맞춰져 있음.

state: question 의 tokens, 지금까지의 output tokens 을 concat

action: vocabulary 에서 다음 token o_i^t 를 얻음. (i-th output 의 t-th token)

- DeepSeekMath는 무엇인가?
- Base Model → Instruct Model (훈련 과정, RL 이전 reference model)
- PPO → GRPO (RL 알고리즘)
- Reward Model - Outcome or Process
- DAPO, Dr.GRPO (GRPO 개선)
- 제가 진행 중인 프로젝트 소개

▼ DeepSeekMath



Pushing the Limits of Mathematical Reasoning in Open Language Models

Language Model 에서의 Mathematical reasoning 어려움!

DeepSeek-Coder-Base-v1.5 7B → Supervised Fine-Tuning → RL → **DeepSeekMath 7B**

1. **Math Corpus 제작:** chain-of-thought (CoT), program-of-thought (PoT), tool-integrated (tool use) reasoning data
2. Group Relative Policy Optimization (GRPO)
 - a. PPO의 변형
 - b. critic model 버림
 - c. baseline? group meanb, c 로 training resource 줄임.

대체로 성능이 좋았다!

▼ Base Model: Math Pre-Training



Data Collection & Training

학습 가능한 Crawler를 사용하여 데이터를 많이 모으고 검증, pre-training에 사용.

DeepSeekMath-Base 7B 모델 제작.

Step-by-step reasoning, few-shot CoT prompting 했을 때 성능 우위

Model	Size	English Benchmarks					Chinese Benchmarks		
		GSM8K	MATH	OCW	SAT	MMLU STEM	CMATH	Gaokao MathCloze	Gaokao MathQA
Closed-Source Base Model									
Minerva	7B	16.2%	14.1%	7.7%	-	35.6%	-	-	-
Minerva	62B	52.4%	27.6%	12.0%	-	53.9%	-	-	-
Minerva	540B	58.8%	33.6%	17.6%	-	63.9%	-	-	-
Open-Source Base Model									
Mistral	7B	40.3%	14.3%	9.2%	71.9%	51.1%	44.9%	5.1%	23.4%
Llemma	7B	37.4%	18.1%	6.3%	59.4%	43.1%	43.4%	11.9%	23.6%
Llemma	34B	54.0%	25.3%	10.3%	71.9%	52.9%	56.1%	11.9%	26.2%
DeepSeekMath-Base	7B	64.2%	36.2%	15.4%	84.4%	56.5%	71.7%	20.3%	35.3%

▼ Instruct Model: Supervised Fine-Tuning



수학 문제에 맞게 답할 수 있도록 Training

영어, 중국어 수학 문제들.

CoT, PoT, tool use 등과 결합하여 **reasoning format** 에 맞춰 776K 개의 examples 제작.

DeepSeekMath-Instruct 7B 모델 제작.

- step-by-step reasoning 을 잘 했다.
- competition-level MATH dataset 에서 잘 했다.
- DeepSeek-LLM-Chat 67B 와 견줄만큼 잘 했다.

▼ PPO



Proximal policy optimization

💡 IDEA

- Clip: update 때 이전의 것과 너무 멀어지지 않게.
- KL Divergence: reference 모델과 너무 멀어지지 않게. (SFT 등을 기준으로)

$$\text{TRPO} - \max_{\theta} \mathbb{E}_t \left[\frac{\pi_{\theta}(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} \hat{A}_t - \beta \text{KL} [\pi_{\theta_{\text{old}}}(\cdot|s_t), \pi_{\theta}(\cdot|s_t)] \right]$$

$$\text{PPO} - \max_{\theta} \mathbb{E}_t \left[\min \left(\frac{\pi_{\theta}(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)} \hat{A}_t, \underbrace{\text{clip} \left(\frac{\pi_{\theta}(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}, 1 - \epsilon, 1 + \epsilon \right)}_{\text{clipping}} \hat{A}_t \right) \right]$$

PPO in LLM

$$\mathcal{J}_{\text{PPO}}(\theta) = \mathbb{E}_{\substack{q \sim P(Q), \\ o \sim \pi_{\theta_{\text{old}}}(O|q)}} \frac{1}{|o|} \sum_{t=1}^{|o|} \min \left[\frac{\pi_{\theta}(o_t|q, o_{<t})}{\pi_{\theta_{\text{old}}}(o_t|q, o_{<t})} A_t, \text{clip} \left(\frac{\pi_{\theta}(o_t|q, o_{<t})}{\pi_{\theta_{\text{old}}}(o_t|q, o_{<t})}, 1 - \epsilon, 1 + \epsilon \right) \right].$$

- q : question 에서
- o : output 을 생성 (language tokens)
- A_t : advantage. Generalized Advantage Estimation (GAE) 적용해 reward 생성

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

- r_{φ} : reward model
- π_{ref} : reference model (initial SFT model)



PPO Value Function

Policy Model 만큼 큼.

RL 에서 value function \Rightarrow baseline, variance 감소에 사용

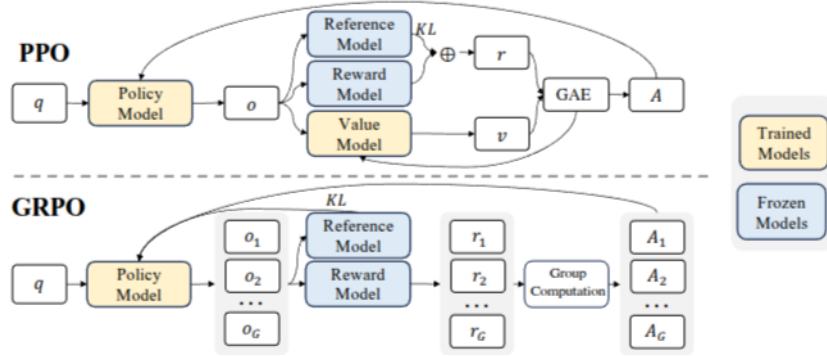
LLM 에서 last token에만 reward 주어짐. 각 token에 대한 정확한 value function 은 복잡함.

▼ GRPO



Group Relative Policy Optimization

GRPO eliminates the value function and estimates the advantage in a group-relative manner.



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

$$\underbrace{\frac{1}{G} \sum_{i=1}^G \frac{1}{|o_i|} \sum_{t=1}^{|o_i|}}_{\text{Group}} \left\{ \underbrace{\mathcal{C}_{\epsilon} \left(\frac{\pi_{\theta}(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}|q, o_{i,<t})}, \hat{A}_{i,t} \right)}_{\text{Clipping}} - \beta D_{KL}[\pi_{\theta} \parallel \pi_{\text{ref}}] \right\}$$

where

$$\mathcal{C}_{\epsilon}(\rho, A) := \min(\rho A, \text{clip}(\rho, 1 - \epsilon, 1 + \epsilon)A)$$

$$D_{KL}[\pi_{\theta} \parallel \pi_{\text{ref}}] = \frac{\pi_{\text{ref}}(o_{i,t}|q, o_{i,<t})}{\pi_{\theta}(o_{i,t}|q, o_{i,<t})} - \log \frac{\pi_{\text{ref}}(o_{i,t}|q, o_{i,<t})}{\pi_{\theta}(o_{i,t}|q, o_{i,<t})} - 1$$

Advantage 계산은 뒤에 이어서.

⚠ PPO와의 차이점

- GRPO는 하나의 question 에 대해 group 으로 output 을 얻음.
- 하나의 group 내에서의 차이로 advantage 를 구함 -> critic (value function) 이 필요 없음.



GRPO Algorithm

Reward Model 을 replay 를 통해 학습시킴.

Algorithm 1 Iterative Group Relative Policy Optimization

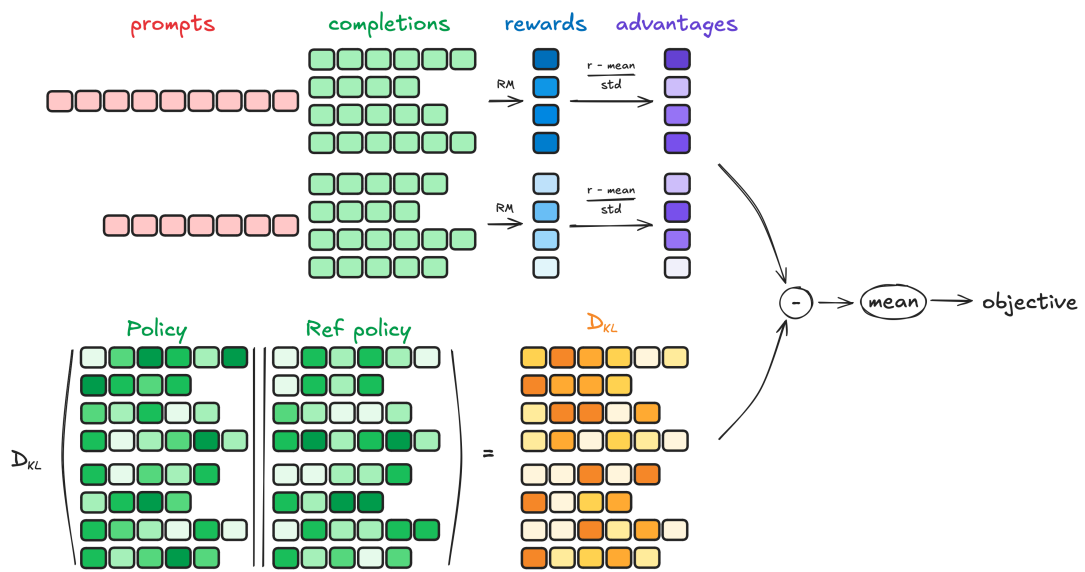
Input initial policy model $\pi_{\theta_{\text{init}}}$; reward models r_{ϕ} ; task prompts \mathcal{D} ; hyperparameters ϵ, β, μ

- 1: policy model $\pi_{\theta} \leftarrow \pi_{\theta_{\text{init}}}$
- 2: **for** iteration = 1, ..., I **do**
- 3: reference model $\pi_{\text{ref}} \leftarrow \pi_{\theta}$
- 4: **for** step = 1, ..., M **do**
- 5: Sample a batch \mathcal{D}_b from \mathcal{D}
- 6: Update the old policy model $\pi_{\theta_{\text{old}}} \leftarrow \pi_{\theta}$
- 7: Sample G outputs $\{o_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot | q)$ for each question $q \in \mathcal{D}_b$
- 8: Compute rewards $\{r_i\}_{i=1}^G$ for each sampled output o_i by running r_{ϕ}
- 9: Compute $\hat{A}_{i,t}$ for the t -th token of o_i through group relative advantage estimation.
- 10: **for** GRPO iteration = 1, ..., μ **do**
- 11: Update the policy model π_{θ} by maximizing the GRPO objective (Equation 21)
- 12: Update r_{ϕ} through continuous training using a replay mechanism.

Output π_{θ}



Diagram



▼ ORM vs. PRM



Reward Model

Reward Model 이 하나의 output 에 대해서 reward 를

한 번만 주는가? → Outcome Supervision RL

step마다 주는가? → Process Supervision RL

reasoning step 기준으로 나눔. LLM token을 사용해서 구분.

각각 ORM, PRM

GRPO + ORM

각 q 에 대해, $\pi_{\theta_{\text{old}}}$ 로부터 G 개의 outputs $\{o_1, o_2, \dots, o_G\}$ 를 생성.

ORM은 G 개의 rewards $\mathbf{r} = \{r_1, r_2, \dots, r_G\}$ 를 생성.

Advantage는 group 평균을 빼고 group 표준편차로 나눠줌.

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

GRPO + PRM

이제 각 output 은 K_i 개의 step 으로 이루어져 있다고 생각.

PRM은 각 output의 step마다 rewards $\mathbf{R} = \{\{r_1^{\text{index}(1)}, \dots, r_1^{\text{index}(K_1)}\}, \dots, \{r_G^{\text{index}(1)}, \dots, r_G^{\text{index}(K_G)}\}\}$ 생성.

$\text{index}(j)$: the end token index of the j-th step

각 output, step 별로 reward 다시 계산. 전체 group 평균을 빼고 group 표준편차로 나눠줌.

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

Advantage 는 이후 얻을 reward 의 합으로 계산

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

▼ DAPO



Token-level Policy Gradient Loss

Decoupled Clip and **D**ynamic **s**Ampling **P**olicy **O**ptimization

Base model: DeepSeek-R1-Zero-Qwen-32B

1. **Removing KL Divergence:** long-CoT reasoning model 에서는 reference model 과 많이 달라질 수 있어서 뺐다.
2. **outcome reward 사용:** 증명, 프로그래밍, 수학 문제 등 정답이 있는 걸로 해서 1, -1 로 reward 설정.

$$\mathcal{J}_{\text{DAPO}}(\theta) = \mathbb{E}_{(q,a) \sim \mathcal{D}, \{o_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot|q)} \left[\underbrace{\frac{1}{\sum_{i=1}^G |o_i|}}_{\text{DAPO}} \sum_{i=1}^G \sum_{t=1}^{|o_i|} \min \left(r_{i,t}(\theta) \hat{A}_{i,t}, \text{clip} \left(r_{i,t}(\theta), \underbrace{1 - \varepsilon_{\text{low}}, 1 + \varepsilon_{\text{high}}}_{\text{Clip-Higher}} \right) \hat{A}_{i,t} \right) \right]$$

where

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

! 차이점

- **Clip-Higher:** $\varepsilon_{\text{high}}$ 의 값을 키움. 확률이 낮은 output 이 $1 + \varepsilon$ 배 확률이 늘어도 여전히 작음. ε_{low} 는 그대로.
- **Dynamic Sampling:** sample 을 더 뽑고 정확도가 1 또는 0이면 제외. 정확도가 1이면 Advantage 역시 0이 되기 때문.
- **Token-Level Policy Gradient Loss:**
 - 기존 식 → 각 sample 별로 평균 후 group 평균. 긴 output 에 포함된 token 은 유용한 답변일 수 있음에도 영향력이 적음.
 - 변경된 식은 긴 output 이 영향력 높음. token 입장에서는 output 길이와 상관 없이 reward 적용 받음.

▼ GRPO

$$\underbrace{\frac{1}{G} \sum_{i=1}^G \frac{1}{|o_i|} \sum_{t=1}^{|o_i|}}_{\text{Group}} \left\{ \underbrace{\mathcal{C}_{\varepsilon} \left(\frac{\pi_{\theta}(o_{i,t}|q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}|q, o_{i,<t})}, \hat{A}_{i,t} \right)}_{\text{Clipping}} - \beta D_{KL}[\pi_{\theta} \parallel \pi_{\text{ref}}] \right\}$$

- **Overlong Reward Shaping:** 패스하겠습니다.

결과: AIME24 avg@32 에 대해서, DAPO 50% vs. DeepSeek-R1-Zero-Qwen-32B 47%
Naive GRPO는 30%



DAPO Algorithm

Algorithm 1 DAPO: Decoupled Clip and Dynamic sAmpling Policy Optimization

Input initial policy model π_θ ; reward model R ; task prompts \mathcal{D} ; hyperparameters $\varepsilon_{\text{low}}, \varepsilon_{\text{high}}$

- 1: **for** step = 1,...,M **do**
- 2: Sample a batch \mathcal{D}_b from \mathcal{D}
- 3: Update the old policy model $\pi_{\theta_{\text{old}}} \leftarrow \pi_\theta$
- 4: Sample G outputs $\{o_i\}_{i=1}^G \sim \pi_{\theta_{\text{old}}}(\cdot|q)$ for each question $q \in \mathcal{D}_b$
- 5: Compute rewards $\{r_i\}_{i=1}^G$ for each sampled output o_i by running R
- 6: Filter out o_i and add the remaining to the dynamic sampling buffer (**Dynamic Sampling** Equation (11))
- 7: **if** buffer size $n_b < N$:
- 8: **continue**
- 9: For each o_i in the buffer, compute $\hat{A}_{i,t}$ for the t -th token of o_i (Equation (9))
- 10: **for** iteration = 1, ..., μ **do**
- 11: Update the policy model π_θ by maximizing the DAPO objective (Equation (8))

Output π_θ

▼ Dr.GRPO



Remove Standard Deviation (std)

Group Relative Policy Optimization **Done Right**

- 우선 DAPO 에서 output 의 길이에 따른 bias 제거 부분을 가져옴.
- 추가로 question 이 어려운 때의 bias 에 대한 문제를 제기.

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

where

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

Advantage 에서 표준편차(std(\mathbf{r}))로 나누던 것을 제거.

▼ GRPO 다시 한 번

$$\underbrace{\frac{1}{G} \sum_{i=1}^G \frac{1}{|o_i|} \sum_{t=1}^{|o_i|}}_{\text{Group}} \left\{ \underbrace{\mathcal{C}_\epsilon \left(\frac{\pi_\theta(o_{i,t}|\mathbf{q}, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}|\mathbf{q}, o_{i,<t})}, \hat{A}_{i,t} \right)}_{\text{Clipping}} - \beta D_{KL}[\pi_\theta \parallel \pi_{\text{ref}}] \right\}$$

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

결과: GRPO 와 성능은 유사. output 길이를 짧게 하는 효과 있었음. (그런데 맞는 답에서의 길이는 그대로고 틀린 답에서 길이만 줄어듦. Overthinking 을 없앴다고 주장.)

▼ 참고 자료

<https://ai-com.tistory.com/entry/RL-강화학습-알고리즘-5-PPO>

DeepSeekMath paper: <https://arxiv.org/abs/2402.03300>

GRPO image: https://huggingface.co/learn/cookbook/fine_tuning_llm_grpo_trl

DAPO paper: <https://arxiv.org/abs/2503.14476>

Dr.GRPO paper: <https://arxiv.org/abs/2503.20783>

GRPO 비유: <https://www.linkedin.com/pulse/%25EC%2589%25BD%25EA%25B2%258C-%25EC%2593%25B0%25EC%2597%25AC%25EC%25A7%2584-grpo-jin-hyung-park-prv4c/>