R1-VL: Learning to Reason with Multimodal Large Language Models via Step-wise Group Relative Policy Optimization

ICCV 2025

Motivation

Limitation of Supervised Fine-Tuning (SFT): Imitation, Not Understanding

- SFT on positive Chain-of-Thought (CoT) data leads models to mimic successful reasoning paths
- However, *models often fail to comprehend what constitutes flawed or incorrect reasoning*

The "Sparse Reward" Issue in Online Reinforcement Learning (RL)

- Recent LLM advancements leverage online RL (e.g., Deepseek-R1's GRPO) with outcome-level rewards
- Challenge for MLLMs: Applying this directly to MLLMs, especially smaller ones, often results in a sparse reward issue

Consequences of Sparse Rewards:

- Poor Efficiency
- Unstable Learning Process

StepGRPO

Step-wise Reasoning Accuracy Reward (StepRAR)

- Rewards the reasoning path using a soft key step matching technique

⇒ **Evaluate whether the reasoning path contains key intermediate reasoning steps**

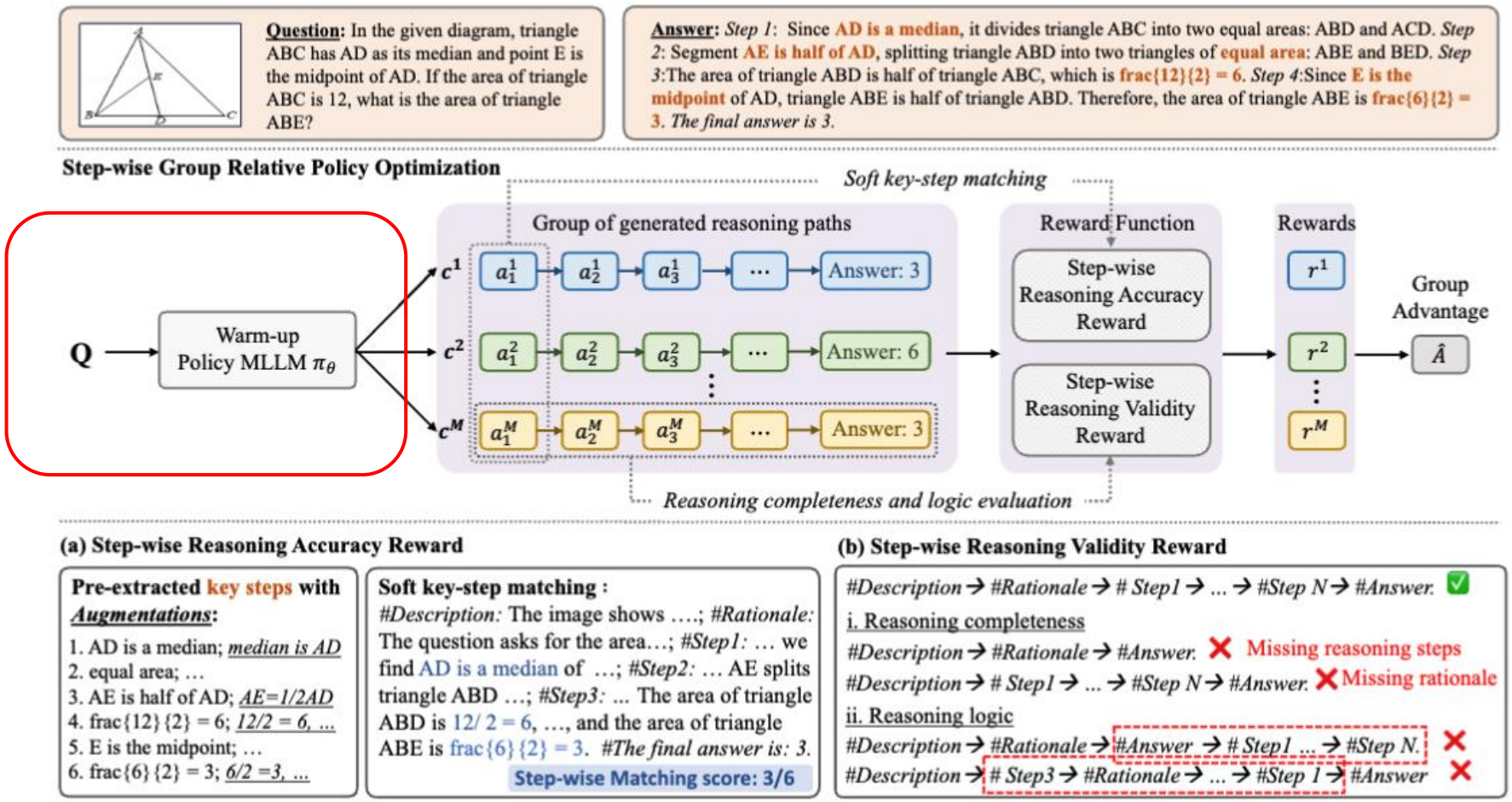
Step-wise Reasoning Validity Reward (StepRVR)

- rewards the reasoning path based on a reasoning completeness and logic evaluation method

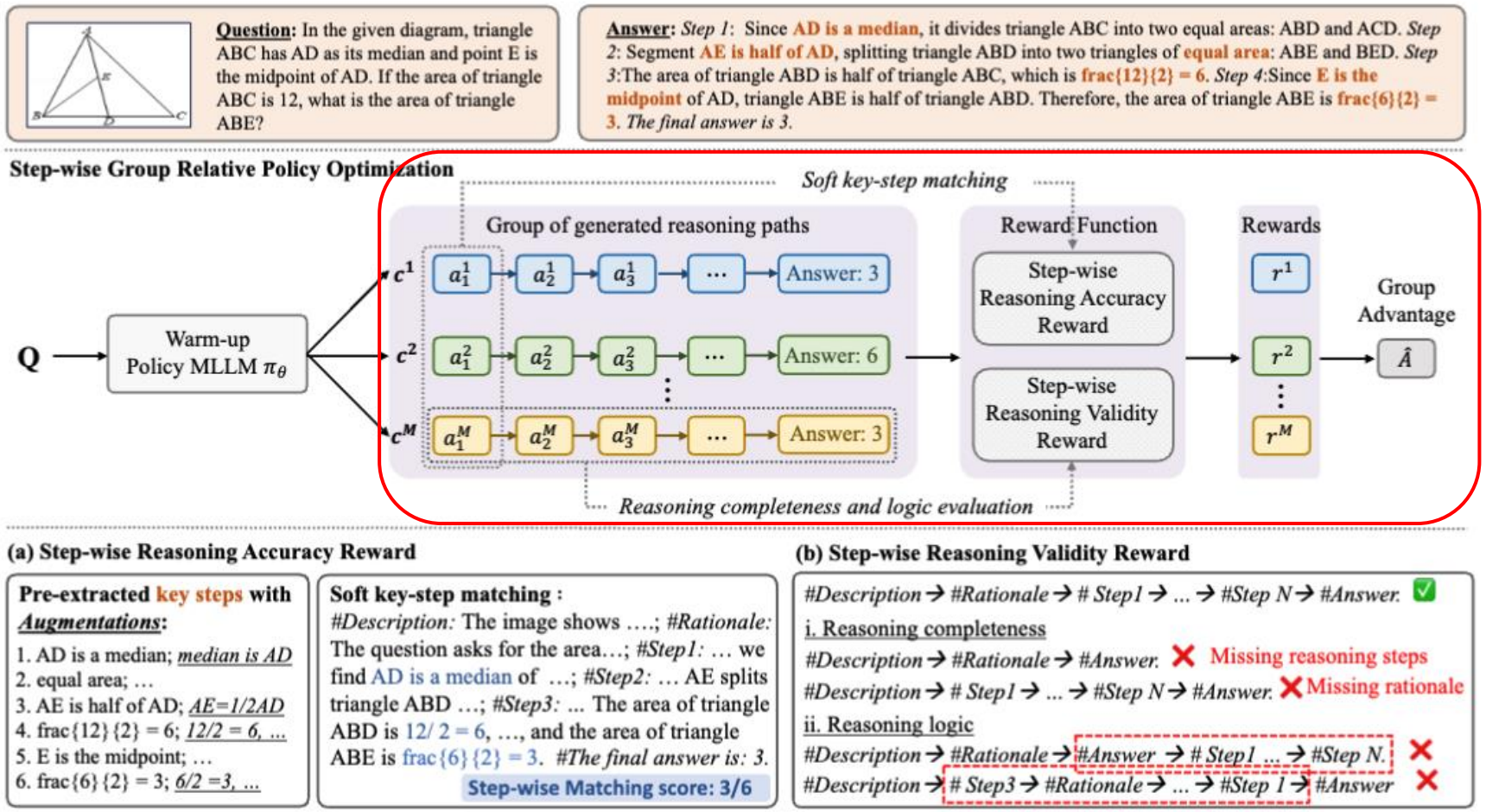
⇒ **Assess whether the reasoning process is well-structured and logically consistent**

Help mitigate the sparse reward issue by providing informative rewards,
even when the reasoning path does not produce the correct final answer

StepGRPO Overview



Overview


Question: In the given diagram, triangle ABC has AD as its median and point E is the midpoint of AD. If the area of triangle ABC is 12, what is the area of triangle ABE?
Answer: Step 1: Since AD is a median, it divides triangle ABC into two equal areas: ABD and ACD. Step 2: Segment AE is half of AD, splitting triangle ABD into two triangles of equal area: ABE and BED. Step 3:The area of triangle ABD is half of triangle ABC, which is frac{12}{2} = 6. Step 4:Since E is the midpoint of AD, triangle ABE is half of triangle ABD. Therefore, the area of triangle ABE is frac{6}{2} = 3. The final answer is 3.
Step-wise Group Relative Policy Optimization
Soft key-step matching
Group of generated reasoning paths
Reward Function
Rewards
Q
Warm-up Policy MLLM π_θ
Answer: 3
Answer: 6
Answer: 3
Step-wise Reasoning Accuracy Reward
Step-wise Reasoning Validity Reward
Group Advantage
Reasoning completeness and logic evaluation
(a) Step-wise Reasoning Accuracy Reward
Pre-extracted key steps with Augmentations:
1. AD is a median; median is AD
2. equal area; …
3. AE is half of AD; AE=1/2AD
4. frac{12}{2} = 6; 12/2 = 6, ...
5. E is the midpoint; …
6. frac{6}{2} = 3; 6/2 =3, ...
Soft key-step matching :
#Description: The image shows …; #Rationale: The question asks for the area…; #Step1: … we find AD is a median of …; #Step2: … AE splits triangle ABD …; #Step3: ... The area of triangle ABD is 12/ 2 = 6, …, and the area of triangle ABE is frac{6}{2} = 3. #The final answer is: 3.
Step-wise Matching score: 3/6
(b) Step-wise Reasoning Validity Reward
#Description → #Rationale → # Step1 → ... → #Step N → #Answer.
i. Reasoning completeness
#Description → #Rationale → #Answer. Missing reasoning steps
#Description → # Step1 → ... → #Step N → #Answer. Missing rationale
ii. Reasoning logic
#Description → #Rationale → #Answer → # Step1 ... → #Step N.
#Description → # Step3 → #Rationale → ... → #Step 1 → #Answer

Experimental Setup

- **Base MLLMs:**
  - Qwen2-VL-2B
  - Qwen2-VL-7B
- **Warm-up Dataset:**
  - Mulberry-260k
- **RL Training Data:**
  - Randomly sampled 10K data from Mulberry-260k
- **Evaluation Benchmarks (8 widely-used multimodal benchmarks):**
  => MathVista, MMStar, Math-Vision, ChartQA, DynaMath, HallusionBench, MathVerse, MME
- **RL Parameters:**
  - Group size: 4 rollouts per question
  - Sampling temperature: 1.2
  - Maximum sequence length: 1024
  - Learning rate: KL divergence coefficient: 0.04
  - Hardware: 4 H100-80GB GPUs

Results

Method	MathVista	MMStar	Math-V	ChartQA	DynaMath	HallBench	MathVerse	MME_{sum}	MMReason	AVG
Closed-Source Model										
GPT-4o [15]	63.8	63.9	30.3	85.7	63.7	55.0	39.4	2329	21.1	56.2
Claude-3.5 Sonnet [1]	67.7	62.2	-	90.8	64.8	55.0	-	1920	-	-
Open-Source Model										
Cambrain-1-8B [38]	49.0	-	-	73.3	-	-	-	-	-	-
MM-1.5-7B [51]	47.6	-	-	78.6	-	-	-	1861	-	-
Idefics3-LLaMA3-8B [18]	58.4	55.9	-	74.8	-	-	-	1937	-	-
InternVL2-8B [8]	58.3	61.5	-	83.3	39.7	-	-	2210	-	-
MiniCPM-V-2.6-8B [48]	60.6	57.5	-	-	-	48.1	-	2348	-	-
DeepSeek-VL2-MOE-4.5B [43]	62.8	61.3	-	86.0	-	-	-	2253	11.5	-
Reasoning Model										
LLaVA-CoT-11B [44]	54.8	57.6	-	-	-	47.8	-	-	-	-
LLaVA-Reasoner-8B [55]	50.6	54.0	-	83.0	-	-	-	-	-	-
Insight-V-8B [10]	49.8	57.4	-	77.4	-	-	-	2069	-	-
Mulberry-7B [46]	63.1	61.3	-	83.9	45.1	54.1	-	2396	11.8	-
LlamaV-o1-11B [37]	54.4	59.4	-	-	-	63.5	-	-	-	-
Vision-R1-7B [14]	73.5	-	-	-	-	-	52.4	-	-	-
LMM-R1 [30]	63.2	58.0	26.3	-	-	-	41.5	-	-	-
R1-ShareVL-7B [47]	75.4	67.0	29.5	-	-	-	52.8	-	-	-
Qwen2-VL-2B [41]	43.0	48.0	12.4	73.5	24.9	41.7	19.7	1872	7.7	37.5
R1-VL-2B (Ours)	52.1	49.8	17.1	75.2	29.4	44.0	26.2	2048	8.3	41.6
Qwen2-VL-7B [41]	58.2	60.7	16.3	83.0	42.1	50.6	32.5	2327	11.9	48.7
R1-VL-7B (Ours)	63.5	60.0	24.7	83.9	45.2	54.7	40.0	2376	12.5	52.1
Qwen2.5-VL-7B [2]	68.2	63.9	25.1	87.3	53.2	52.1	49.2	2347	17.3	55.5
R1-VL-7B* (Ours)	74.3	66.2	28.2	87.7	56.5	57.2	52.2	2395	17.9	58.4

Table 1. Main experimental results. To comprehensively examine the proposed StepGRPO, we conduct extensive experiments with two baseline models on eight benchmarks, and compare StepGRPO with various state-of-the-art MLLMs.* indicates that the model is trained using Qwen2.5-VL-7B as the base model with the data from [47].

Warm-up	Step-wise reasoning rewards		MathVista
	StepRAR	StepRVR	
			58.2
✓			61.2
✓	✓		62.4
✓		✓	61.9
✓	✓	✓	**63.5**

Table 2. Ablation study of StepGRPO over Qwen2-VL-7B.

- **R1-VL Achieves SOTA Performance Among Reasoning MLLMs**
- **Competitive Results Against General & Closed-Source MLLMs**
- StepGRPO effectively improves complex reasoning tasks by reinforcing **both the correctness of intermediate steps and the overall logical structure of the reasoning process**

Analysis

	Number of generations M per question				
Method	2	3	4	5	6
R1-VL-7B	62.5	62.8	63.5	63.2	63.7

Table 3. Parameter analysis of M. The experiments are conducted on Qwen2-VL-7B over MathVista.

Method	MathVista
Warm-up	61.7
Warm-up + Outcome-level reward	62.3
Warm-up + Step-wise reward (Ours)	**63.5**

Table 4. Effectiveness of the step-wise reasoning rewards. The experiments are conducted on Qwen2-VL-7B over MathVista.

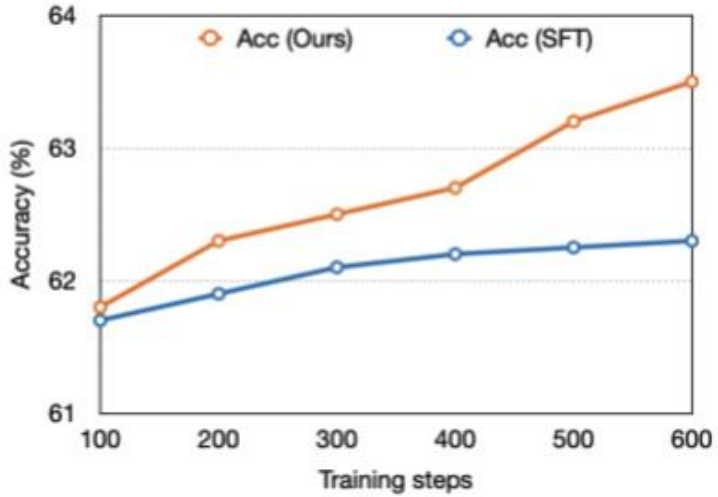


Figure 3. Comparison between StepGRPO and SFT. The experiments are conducted on Qwen2-VL-7B over MathVista.

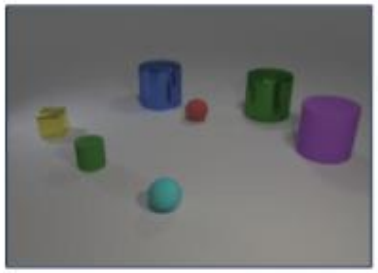
- **A larger M generally leads to better performance**

⟹ A larger M results in a more stable and accurate baseline estimation,
whereas a small M may lead to high variance in baseline estimation, making the optimization process less reliable

- **The importance of process reward**

⟹ Step wise rewards are more effective in enhancing MLLMs' reasoning capabilities, as they provide more fine-grained supervision and largely mitigate the sparse reward issue

Qualitative comparison



Question: Subtract all red things. Subtract all tiny matte balls. How many objects are left?

Correct answer: 5

Qwen2-VL-7B:

First, identify the red objects in the image. There is 1 red object, which is a small red sphere. ✗ Incorrect reasoning

Then, subtract the red object from the total number of objects. There are 7 objects in total and subtracting the 1 red object leaves 6 objects. Identify the tiny matte balls in the image. There are 2 tiny matte balls, which are the small green cylinder and the cyan sphere. Subtract the tiny matte balls from the remaining objects.

After subtracting the red object, there are 6 objects left. Subtracting the 2 tiny matte balls leaves 4 objects. Answer: 4

Mulberry-7B:

Image Description: The image shows several 3D objects on a flat surface. There are two red objects: a small red sphere and a larger green cylinder. Other objects include a yellow cube, a green cylinder, a blue cylinder, a purple cylinder, and a teal sphere.

Rationales: The question asks to subtract all red things and all tiny matte balls. We need to identify these objects in the image and count the remaining ones.

Let's think step by step.

Step 1: Identify and count all red objects. There is one red sphere. ✗ Incorrect reasoning

Step 2: Identify and count all tiny matte balls. There is one teal sphere, which is not red, so it is not subtracted.

Step 3: Subtract the red objects and tiny matte balls from the total number of objects: 7 - 1 = 6.

The final answer is: 6

R1-VL-7B (Ours):

Image Description:
The image shows a collection of 3D geometric objects on a flat surface. There are various colored shapes, including a red sphere, a yellow cube, a green cylinder, a blue cylinder, a purple cylinder, and a cyan sphere.

Rationales:
The question asks to subtract all red things and all tiny matte balls, then determine how many objects are left. This involves identifying and counting specific objects based on their color and size.

Let's think step by step.

Step 1:
Identify and count all objects in the image. There are seven objects: a yellow cube, a green cylinder, a blue cylinder, a purple cylinder, a red sphere, and a cyan sphere. ✓

Step 2:
Subtract all red things. There is one red sphere. ✓

Step 3:
Subtract all tiny matte balls from the remaining objects. There is one cyan sphere, which is matte and small. ✓

Step 4:
Count the remaining objects after removing the red sphere and the cyan sphere. ✓

The final answer is: 5 ✓

Figure 4. Qualitative comparison.

SimPO: Simple Preference Optimization with a Reference-Free Reward

NeurIPS 2024

Preliminary: Direct Preference Optimization (DPO)

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta\log\frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}\right)\right].$$

win, loss, probability of y_w, probability of y_l

- **Basic Regime:**

단일 쿼리에 대해 올바른 답변(win)과 잘못된 답변(lose)의 차이(margin)을 극대화

⟹ **y_win을 생성할 확률을 상대적으로 높이고, y_lose를 생성할 확률을 상대적으로 낮춤**

- **Reward Function:**

보상 함수는 Reference 모델과의 확률 비율을 사용

- **로그 확률 계산 시, token 별 probability를 '합계' 하여 문장을 생성할 probability를 계산**

```
all_logps = per_token_logps[:, 1:].sum(-1)
```

Motivation

Training-Inference Discrepancy

- 모델은 추론 시 확률 평균으로 생성하지만, 학습은 상대적 확률 비율로 최적화

⟹ 기존 DPO의 보상 함수는 Reference 모델과의 확률 비율을 사용하지만, 실제 추론 시 모델은 '평균 Log-Likelihood'에 의존

```
if average_log_prob:
    return (per_token_logps * loss_mask).sum(-1) / loss_mask.sum(-1)
```

Reference-Free Efficiency

- 불필요하게 메모리를 점유하는 Reference 모델의 존재

⟹ SimPO는 Reference 모델을 완전히 제거함으로써 계산 자원을 약 20% 절감

Length Bias & Margin Optimization

- 단순히 긴 답변이 아닌, 품질 차이에 집중하는 마진(Margin)이 필요

⟹ 보상 함수에 길이 정규화(Length Normalization)를 명시적으로 도입하여 '길기만 한 답변'이 고득점을 받는 현상을 방지

SimPO

$$\mathcal{L}_{\text{DPO}}(\pi_\theta; \pi_{\text{ref}}) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(\beta\log\frac{\pi_\theta(y_w \mid x)}{\pi_{\text{ref}}(y_w \mid x)} - \beta\log\frac{\pi_\theta(y_l \mid x)}{\pi_{\text{ref}}(y_l \mid x)}\right)\right]$$

$$\mathcal{L}_{\text{SimPO}}(\pi_\theta) = -\mathbb{E}_{(x,y_w,y_l)\sim\mathcal{D}}\left[\log\sigma\left(\frac{\beta}{|y_w|}\log\pi_\theta(y_w|x) - \frac{\beta}{|y_l|}\log\pi_\theta(y_l|x) - \gamma\right)\right]$$

1. 모델이 y_win을 생성할 확률과 y_lose를 생성할 확률의 차이를 직접적으로 극대화

2. 평균 생성확률의 도입

3. Margin의 도입

참조 모델의 필요성을 제거하고 생성 메트릭과 직접적으로 일치하는 암시적 보상 형식을 사용

Setup

Table 2: Evaluation details for AlpacaEval 2 [55], Arena-Hard [54], and MT-Bench [99]. The baseline model refers to the model compared against. GPT-4 Turbo corresponds to GPT-4-Preview-1106.

	# Exs.	Baseline Model	Judge Model	Scoring Type	Metric
AlpacaEval 2	805	GPT-4 Turbo	GPT-4 Turbo	Pairwise comparison	LC & raw win rate
Arena-Hard	500	GPT-4-0314	GPT-4 Turbo	Pairwise comparison	Win rate
MT-Bench	80	-	GPT-4/GPT-4 Turbo	Single-answer grading	Rating of 1-10

Method	Objective
RRHF [91]	$\max\left(0, -\frac{1}{\lvert y_w\rvert}\log \pi_\theta(y_w\lvert x) + \frac{1}{\lvert y_l\rvert}\log \pi_\theta(y_l\lvert x)\right) - \lambda \log \pi_\theta(y_w\lvert x)$
SLiC-HF [96]	$\max\left(0, \delta - \log \pi_\theta(y_w\lvert x) + \log \pi_\theta(y_l\lvert x)\right) - \lambda \log \pi_\theta(y_w\lvert x)$
DPO [66]	$-\log \sigma\left(\beta \log \frac{\pi_\theta(y_w\lvert x)}{\pi_{\text{ref}}(y_w\lvert x)} - \beta \log \frac{\pi_\theta(y_l\lvert x)}{\pi_{\text{ref}}(y_l\lvert x)}\right)$
IPO [6]	$\left(\log \frac{\pi_\theta(y_w\lvert x)}{\pi_{\text{ref}}(y_w\lvert x)} - \log \frac{\pi_\theta(y_l\lvert x)}{\pi_{\text{ref}}(y_l\lvert x)} - \frac{1}{2\tau}\right)^2$
CPO [88]	$-\log \sigma\left(\beta \log \pi_\theta(y_w\lvert x) - \beta \log \pi_\theta(y_l\lvert x)\right) - \lambda \log \pi_\theta(y_w\lvert x)$
KTO [29]	$-\lambda_w \sigma\left(\beta \log \frac{\pi_\theta(y_w\lvert x)}{\pi_{\text{ref}}(y_w\lvert x)} - z_{\text{ref}}\right) + \lambda_l \sigma\left(z_{\text{ref}} - \beta \log \frac{\pi_\theta(y_l\lvert x)}{\pi_{\text{ref}}(y_l\lvert x)}\right)$, where $z_{\text{ref}} = \mathbb{E}_{(x,y)\sim\mathcal{D}}\left[\beta \mathbf{KL}\left(\pi_\theta(y\lvert x)\lVert\pi_{\text{ref}}(y\lvert x)\right)\right]$
ORPO [42]	$-\log p_\theta(y_w\lvert x) - \lambda \log \sigma\left(\log \frac{p_\theta(y_w\lvert x)}{1-p_\theta(y_w\lvert x)} - \log \frac{p_\theta(y_l\lvert x)}{1-p_\theta(y_l\lvert x)}\right)$, where $p_\theta(y\lvert x) = \exp\left(\frac{1}{\lvert y\rvert}\log \pi_\theta(y\lvert x)\right)$
R-DPO [64]	$-\log \sigma\left(\beta \log \frac{\pi_\theta(y_w\lvert x)}{\pi_{\text{ref}}(y_w\lvert x)} - \beta \log \frac{\pi_\theta(y_l\lvert x)}{\pi_{\text{ref}}(y_l\lvert x)} + (\alpha\lvert y_w\rvert - \alpha\lvert y_l\rvert)\right)$
SimPO	$-\log \sigma\left(\frac{\beta}{\lvert y_w\rvert}\log \pi_\theta(y_w\lvert x) - \frac{\beta}{\lvert y_l\rvert}\log \pi_\theta(y_l\lvert x) - \gamma\right)$

Model 구성

- Mistral / Llama3 Base & Instruct사용
- Base의 경우 UltraChat200k 로 SFT된 모델에서 출발
- Instruct의 경우 시판된 off-the-shelf 모델 사용

Training Data 구성

- SFT 모델로 5개 응답 생성
- RM을 기반으로 reward를 측정 후,
  높은 샘플을 chosen, 낮은 샘플을 reject으로 구성

⟹ on-policy 설정에 더 가깝도록

Result

Table 4: AlpacaEval 2 [55], Arena-Hard [54], and MT-Bench [99] results under the four settings. LC and WR denote length-controlled and raw win rate, respectively. We train SFT models for Base settings on the UltraChat dataset. For Instruct settings, we use off-the-shelf models as the SFT model.

Method	Mistral-Base (7B)					Mistral-Instruct (7B)				
	AlpacaEval 2		Arena-Hard	MT-Bench		AlpacaEval 2		Arena-Hard	MT-Bench	
	LC (%)	WR (%)	WR (%)	GPT-4 Turbo	GPT-4	LC (%)	WR (%)	WR (%)	GPT-4 Turbo	GPT-4
SFT	8.4	6.2	1.3	4.8	6.3	17.1	14.7	12.6	6.2	7.5
RRHF [91]	11.6	10.2	5.8	5.4	6.7	25.3	24.8	18.1	6.5	7.6
SLiC-HF [96]	10.9	8.9	7.3	5.8	**7.4**	24.1	24.6	18.9	6.5	**7.8**
DPO [66]	15.1	12.5	10.4	5.9	7.3	26.8	24.9	16.3	6.3	7.6
IPO [6]	11.8	9.4	7.5	5.5	7.2	20.3	20.3	16.2	6.4	**7.8**
CPO [88]	9.8	8.9	6.9	5.4	6.8	23.8	28.8	**22.6**	6.3	7.5
KTO [29]	13.1	9.1	5.6	5.4	7.0	24.5	23.6	17.9	6.4	7.7
ORPO [42]	14.7	12.2	7.0	5.8	7.3	24.5	24.9	20.8	6.4	7.7
R-DPO [64]	17.4	12.8	8.0	5.9	**7.4**	27.3	24.5	16.1	6.2	7.5
SimPO	**21.5**	**20.8**	**16.6**	**6.0**	7.3	**32.1**	**34.8**	21.0	**6.6**	7.6
Method	**Llama-3-Base (8B)**					**Llama-3-Instruct (8B)**				
	AlpacaEval 2		Arena-Hard	MT-Bench		AlpacaEval 2		Arena-Hard	MT-Bench	
	LC (%)	WR (%)	WR (%)	GPT-4 Turbo	GPT-4	LC (%)	WR (%)	WR (%)	GPT-4 Turbo	GPT-4
SFT	6.2	4.6	3.3	5.2	6.6	26.0	25.3	22.3	6.9	8.1
RRHF [91]	12.1	10.1	6.3	5.8	7.0	31.3	28.4	26.5	6.7	7.9
SLiC-HF [96]	12.3	13.7	6.0	6.3	7.6	26.9	27.5	26.2	6.8	8.1
DPO [66]	18.2	15.5	15.9	6.5	7.7	40.3	37.9	32.6	**7.0**	8.0
IPO [6]	14.4	14.2	17.8	6.5	7.4	35.6	35.6	30.5	**7.0**	**8.3**
CPO [88]	10.8	8.1	5.8	6.0	7.4	28.9	32.2	28.8	**7.0**	8.0
KTO [29]	14.2	12.4	12.5	6.3	**7.8**	33.1	31.8	26.4	6.9	8.2
ORPO [42]	12.2	10.6	10.8	6.1	7.6	28.5	27.4	25.8	6.8	8.0
R-DPO [64]	17.6	14.4	17.2	**6.6**	7.5	41.1	37.8	33.1	**7.0**	8.0
SimPO	**22.0**	**20.3**	**23.4**	**6.6**	7.7	**44.7**	**40.5**	**33.8**	**7.0**	8.0

- **SimPO는 모델 크기나 학습 데이터 설정에 상관없이 모든 벤치마크에서 높은 성능**

⇒ MT-Bench는 모델 간 변별력이 낮았지만, 더 변별력이 높은 Arena-Hard에서 SimPO가 일관되게 높은 승률

- **단순한 변형을 넘어, 기존의 대세인 DPO를 유의미한 차이로 앞섬**

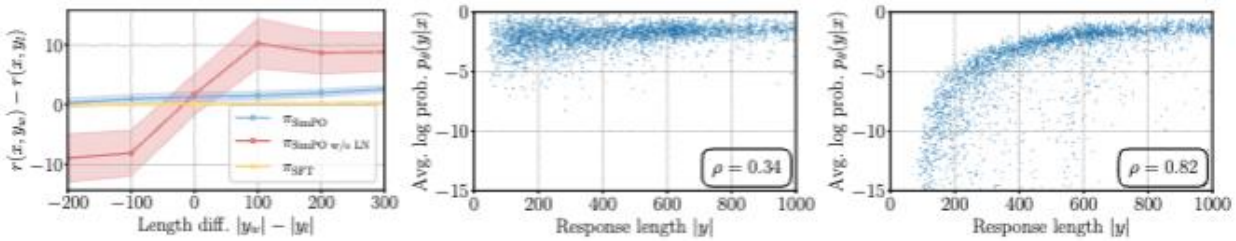
⇒ Reference 모델 없이도 더 정교한 정렬이 가능

- **이미 강력한 모델(Instruct-tuned)을 더욱 날카롭게 다듬는 데 가장 효과적**

Ablation

Table 5: Ablation studies under Mistral-Base and Mistral-Instruct settings. We ablate each key design of SimPO: (1) removing length normalization in Eq. (4) (*i.e.*, w/o LN); (2) setting target reward margin γ to be 0 in Eq. (6) (*i.e.*, $\gamma = 0$).

Method	Mistral-Base (7B) Setting					Mistral-Instruct (7B) Setting				
	AlpacaEval 2		Arena-Hard	MT-Bench		AlpacaEval 2		Arena-Hard	MT-Bench	
	LC (%)	WR (%)	WR (%)	GPT-4 Turbo	GPT-4	LC (%)	WR (%)	WR (%)	GPT-4 Turbo	GPT-4
DPO	15.1	12.5	10.4	5.9	7.3	26.8	24.9	16.3	6.3	7.6
SimPO	21.5	20.8	16.6	6.0	7.3	32.1	34.8	21.0	6.6	7.6
w/o LN	11.9	13.2	9.4	5.5	7.3	19.1	19.7	16.3	6.4	7.6
$\gamma = 0$	16.8	14.3	11.7	5.6	6.9	30.9	34.2	20.5	6.6	7.7



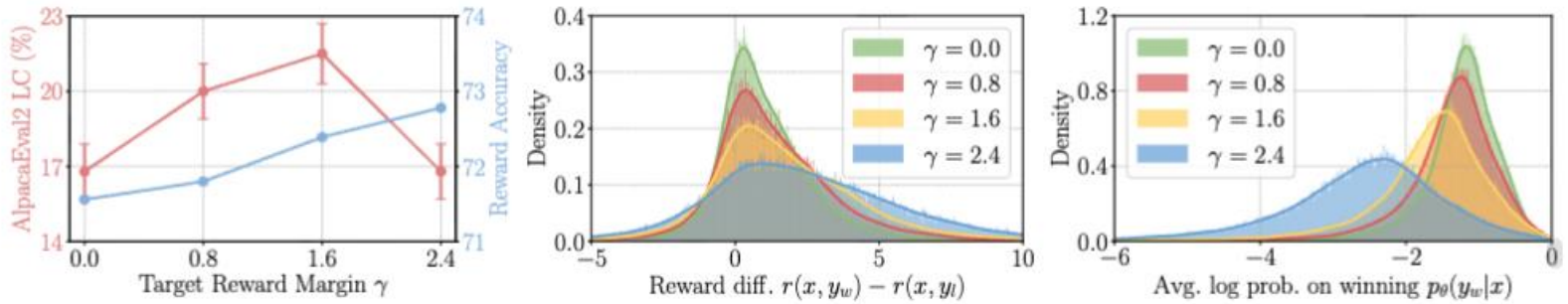
(a) Reward optimization. (b) SimPO. (c) SimPO without LN.

- **LN이 적용된 SimPO는 응답 쌍의 길이 차이에 관계없이 모든 응답 쌍에 대해 일관되게 양의 보상 마진을 달성**

- LN이 없는 SimPO는 승리 응답이 패배 응답보다 짧을 때 선호 쌍에 대해 음의 보상 차이를 유발

⇒ 이는 모델이 제대로 학습하지 못함을 의미

The Impact of Target Reward Margin in SimPO



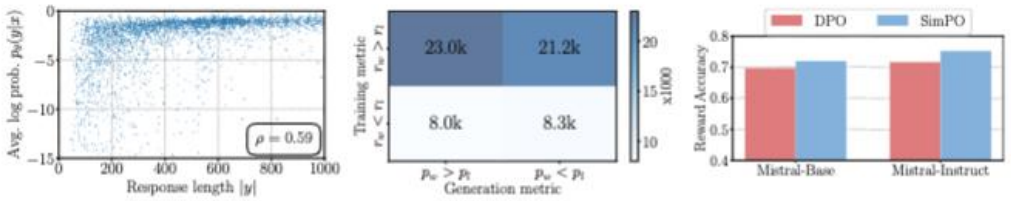
(a) Performance w/ different γ. (b) Reward diff. distribution. (c) Log prob. distribution.

Figure 3: Study of the margin γ. (a) Reward accuracy and AlpacaEval2 LC win rate under different γ values. (b) Reward difference distribution under different γ values. (c) Log likelihood distribution on chosen responses under different γ values.

- **마진증가에 따른 보상 정확도(Reward Accuracy)의 비례 상승**
- **응답 품질과 마진 사이의 최적점(Trade-off) 존재**

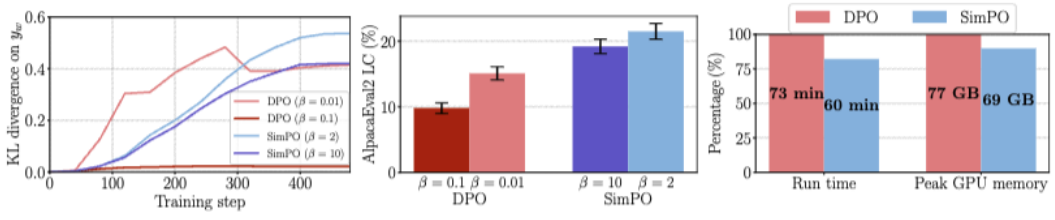
⟹ 적절한 마진은 보상 분포를 넓게 퍼뜨려 모델의 변별력을 높임

In-Depth Analysis of DPO vs. SimPO



(a) Length correlation (DPO). (b) Contingency table (DPO). (c) Reward Accuracy.

Figure 4: Comparison between SimPO and DPO, measured on UltraFeedback. (a) Spearman correlation between average log probability and response length for DPO. (b) Contingency table of rankings based on DPO rewards and the average log likelihood (measured on the training set). (c) Reward accuracy of DPO and SimPO.



(a) KL divergence w/ different β. (b) Performance w/ different β. (c) Efficiency of DPO vs. SimPO.

Figure 5: Comparison between SimPO and DPO (continued). (a) With different β in DPO and SimPO, KL divergence from the policy model to the reference model on y_w. (b) AlpacaEval2 LC win rate of DPO and SimPO with different β. (c) Runtime and memory usage for DPO and SimPO.

- **DPO 보상 체계와 실제 생성 확률 사이의 심각한 괴리**

⟹ DPO에서 보상이 높다고 해서,
모델이 그 답변을 실제로 생성할 확률이 높은 것은 아님

- **DPO는 Reference 모델을 통해 간접적으로 길이를 조절하려 하지만, 한계가 명확**
- **단순한 보상 함수가 오히려 데이터의 선호 관계를 더 정확하게 학습**

- **Reference 모델을 통한 명시적 제약 없이도 모델은 안정적으로 학습**
- 베타의 적절한 강도의 최적화가 성능을 결정
- **Reference 모델 제거는 실제 학습 시간과 메모리의 획기적인 절감**

Thank you