VAPO & DAPO
An Open-Source LLM RL System at Scale

ByteDance Seed & Tsinghua AIR

Outline

- Background
- DAPO: SOTA LLM RL algorithm w/o critic
- VAPO: SOTA LLM RL algorithm w/ critic

Background

- O1-like reasoning ability for math-problem solving

Question:
Given that the base ABC of the tetrahedron $S-ABC$ is an equilateral triangle, the projection H of point A on the face SBC is the orthocenter of $\triangle SBC$, the dihedral angle $H-AB-C$ is $30°$, and $SA=2$, find the volume of the tetrahedron. The answer is in the form $\frac{k}{m}$, so give the value of $k+m$.

Response:
Let us consider the given problem: We need to determine ...
...
Next, let S have coordinates (x_0, y_0, z_0). Since $SA = 2$, the distance formula gives us $\sqrt{(x_0)^2+(y_0)^2+(z_0)^2} = 2$, so we can write $x_0^2+y_0^2+z_0^2 = 4$. Now, remember that H is the orthogonal projection of
...
However, wait a moment, let's rethink about the dihedral angle involving planes in a more thoughtful geometric way. Consider the plane $\alpha_1 = ABC$, which lies entirely on the xy coordinate plane (as $z=0$). Let the plane $\alpha_2 = SBC$. The point A projected perpendicularly to plane α_2 lands on H. The line $l = AB$ ...
...

Table 2 Emergence of Reflective Behavior in Reinforcement Learning

RL Background

- The objective of PPO:
  - $\hat{A}_t$: advantage of each action. Critic model is used for estimation.
  - Importance sampling, to correct the expectation
  - Clipping, for updating within the trust-region

$$\mathcal{J}_{\mathrm{PPO}}(\theta) = \mathbb{E}_{(q,a)\sim\mathcal{D}, o_{\le t}\sim\pi_{\theta_{\mathrm{old}}}(\cdot|q)} \left[\min\left(\frac{\pi_\theta(o_t \mid q, o_{<t})}{\pi_{\theta_{\mathrm{old}}}(o_t \mid q, o_{<t})} \boxed{\hat{A}_t}, \boxed{\mathrm{clip}} \left(\boxed{\frac{\pi_\theta(o_t \mid q, o_{<t})}{\pi_{\theta_{\mathrm{old}}}(o_t \mid q, o_{<t})}}, 1-\varepsilon, 1+\varepsilon \right) \hat{A}_t \right) \right], \quad (1)$$

RL Background

- GRPO differs in the advantage estimation
  - Sharing the same loss objective with PPO
  - $\hat{A}_t$: estimated in a group-relative manner

$$\hat{A}_{i,t} = \frac{r_i - \mathrm{mean}(\{R_i\}_{i=1}^G)}{\mathrm{std}(\{R_i\}_{i=1}^G)}$$

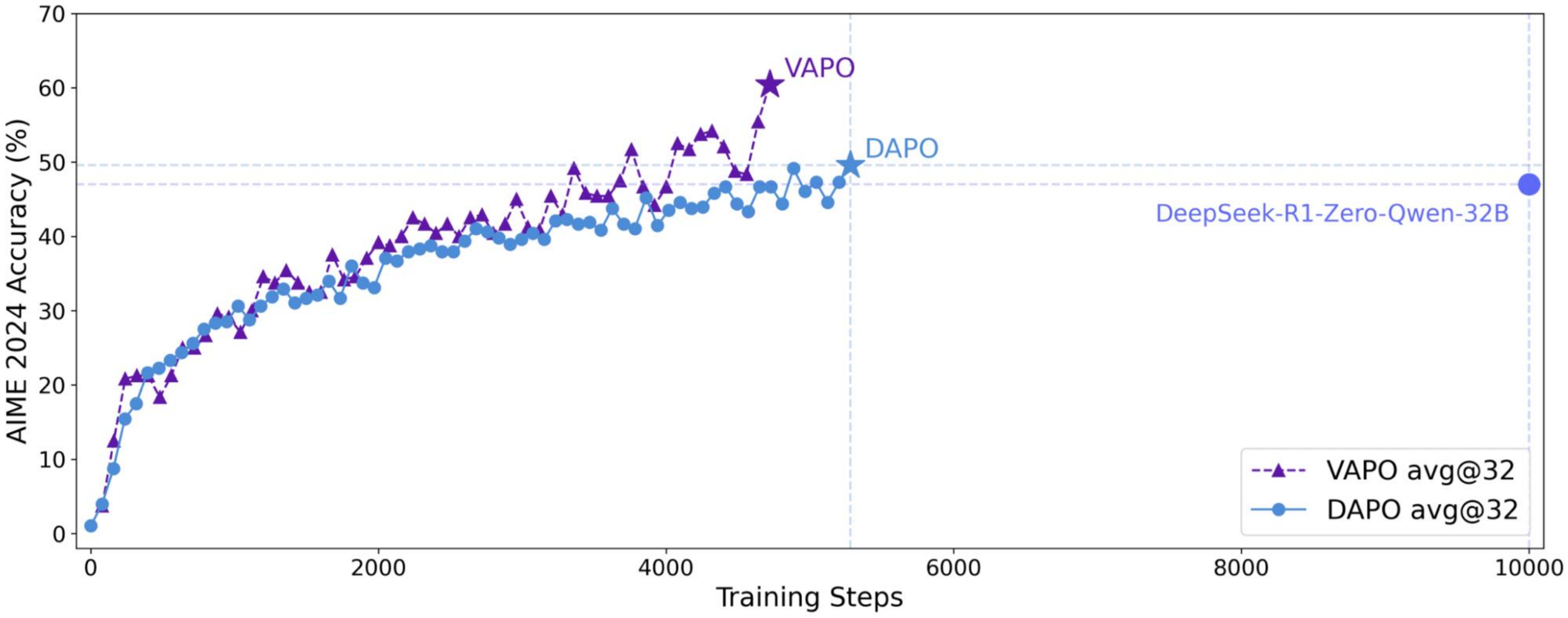
  - Trajectory-level loss

$$\mathcal{J}_{\mathrm{GRPO}}(\theta) = \mathbb{E}_{(q,a)\sim\mathcal{D},\{o_i\}_{i=1}^G\sim\pi_{\theta_{\mathrm{old}}}(\cdot|q)}$$
$$\left[\boxed{\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}}\left(\min\left(r_{i,t}(\theta)\hat{A}_{i,t},\ \mathrm{clip}\Big(r_{i,t}(\theta), 1-\varepsilon, 1+\varepsilon\Big)\hat{A}_{i,t}\right) - \beta D_{\mathrm{KL}}(\pi_\theta||\pi_{\mathrm{ref}})\right)\right]$$

Performance

Zero-setting: RL from the pretrained model, established by DeepSeek-R1

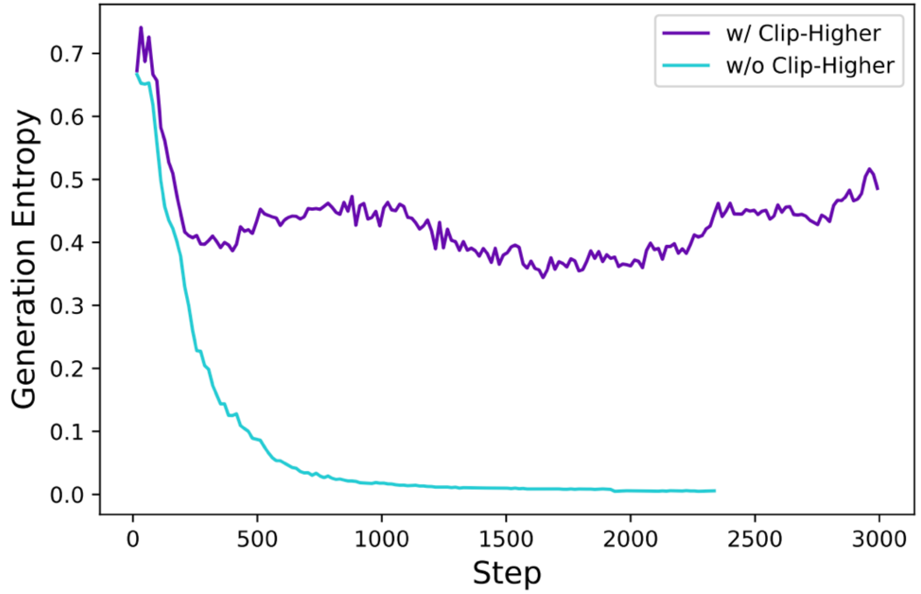
- VAPO: SOTA of algorithms w/ critic
- DAPO: SOTA of algorithms w/o critic



DAPO

Clip-Higher

- Problem: entropy collapse
  - Similar generations under the same prompt
  - Less exploration
- We have tried: temperature / topp, entropy loss, eps-greedy sampling



(b) Entropy of actor model.

Clip-Higher

- We find ϵ_{high} bounds the entropy heavily.
- An informal intuition:
  - If π_{old}= 0.9, ϵ = 0.2, the upper bound is 0.9 * 1.2 = **1.08** (donot work)
  - If π_{old} = 0.01 , ϵ = 0.2, the upper bound is 0.01 * 1.2 = **0.012** (it works)

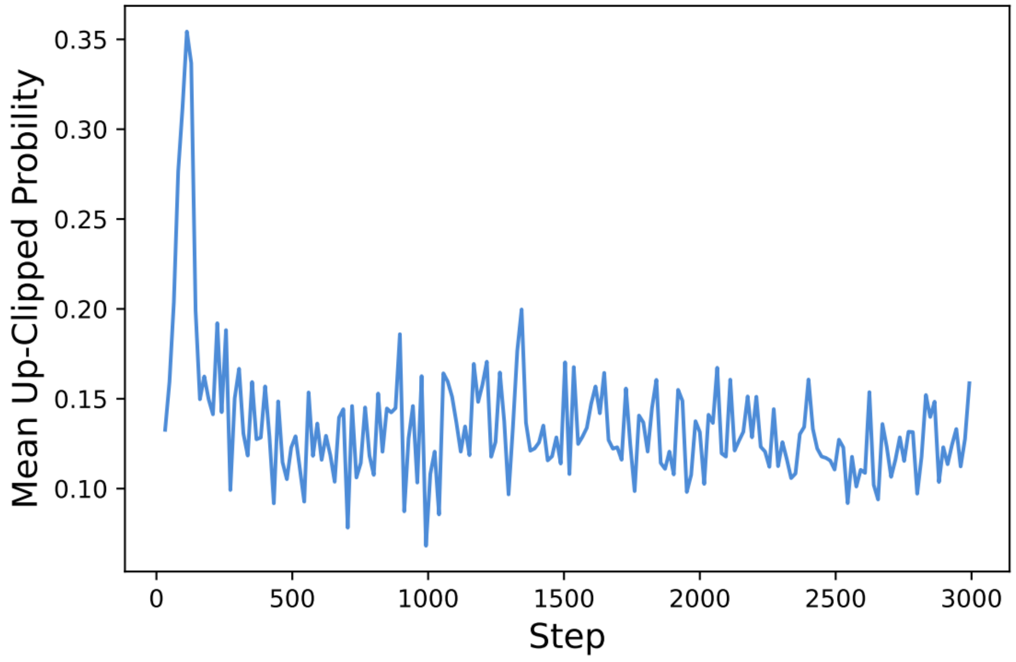
$$
\begin{aligned}
\mathcal{J}_{\text{DAPO}}(\theta) = \quad & \mathbb{E}_{(q,a)\sim\mathcal{D},\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|q)} \\
& \left[\frac{1}{\sum_{i=1}^{G}|o_i|}\sum_{i=1}^{G}\sum_{t=1}^{|o_i|}\min\Big(r_{i,t}(\theta)\hat{A}_{i,t},\ \text{clip}\Big(r_{i,t}(\theta), 1-\varepsilon_{\text{low}}, 1+\boxed{\varepsilon_{\text{high}}}\Big)\hat{A}_{i,t}\Big)\right] \\
\text{s.t.} \quad & 0 < \Big|\{o_i \mid \texttt{is\_equivalent}(a, o_i)\}\Big| < G,
\end{aligned}
\tag{8}
$$

where

$$
r_{i,t}(\theta) = \frac{\pi_\theta(o_{i,t} \mid q, o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q, o_{i,<t})}, \quad \hat{A}_{i,t} = \frac{R_i - \text{mean}(\{R_i\}_{i=1}^{G})}{\text{std}(\{R_i\}_{i=1}^{G})}. \tag{9}
$$

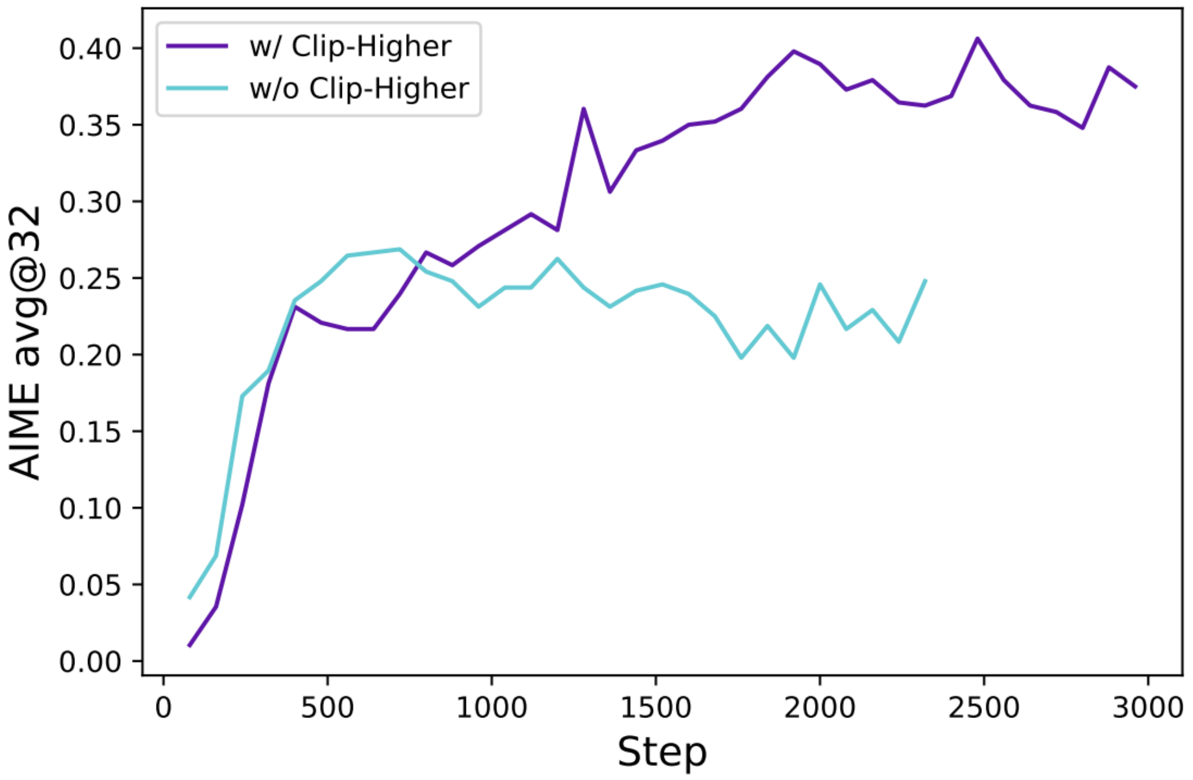
Clip-Higher

- An informal intuition:
  - If π_{old} = 0.9, ϵ = 0.2, the upper bound is 0.9 $*$ 1.2 = **1.08** (donot work)
  - If π_{old} = 0.01 , ϵ = 0.2, the upper bound is 0.01 $*$ 1.2 = **0.012** (it works)

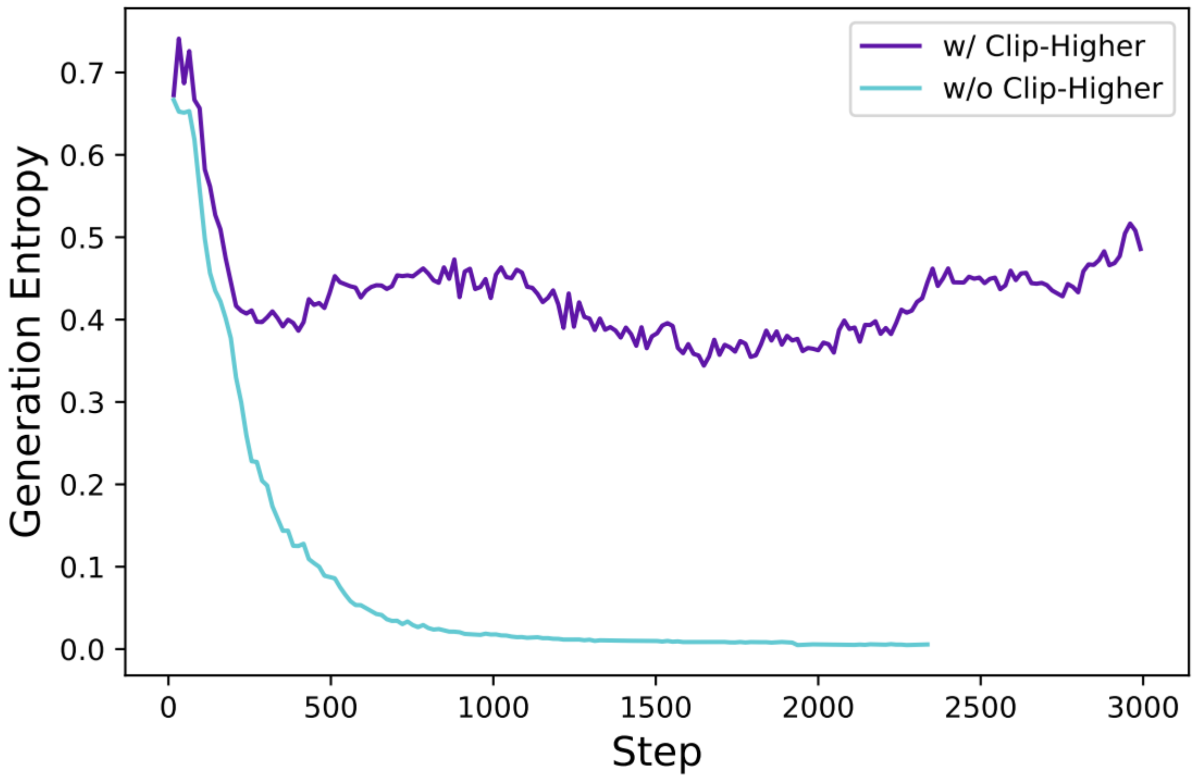


(a) Maximum clipped probabilities.

Clip-Higher



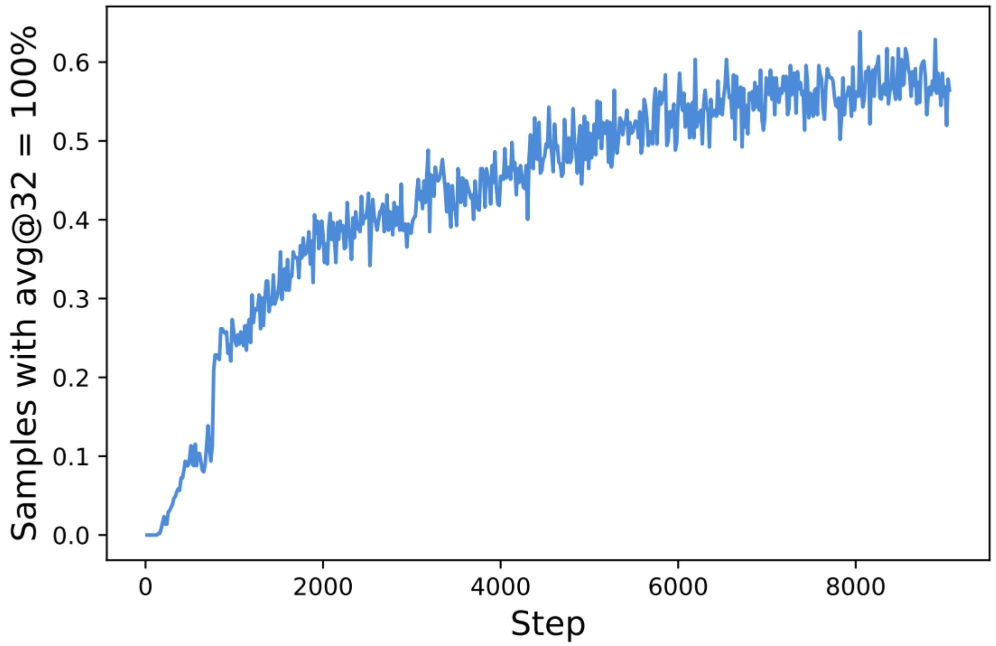
(a) Accuracies on AIME.



(b) Entropy of actor model.

Dynamic Sampling

- The number of effective prompts shrink rapidly, e.g. 512 -> 150
- Effects:
  - Larger gradient variance
    - Make the training unstable
  - Lower gradient norm
    - Slow down training



(b) The proportion of samples with an accuracy of 1.

Dynamic Sampling

- Method: keep sampling until the batch is fulfilled with effective prompts

$$
\begin{aligned}
\mathcal{J}_{\mathrm{DAPO}}(\theta) = \quad & \mathbb{E}_{(q,a)\sim\mathcal{D},\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\mathrm{old}}}(\cdot|q)} \\
& \left[\frac{1}{\sum_{i=1}^{G}|o_i|}\sum_{i=1}^{G}\sum_{t=1}^{|o_i|}\min\left(r_{i,t}(\theta)\hat{A}_{i,t},\ \mathrm{clip}\Big(r_{i,t}(\theta), 1-\varepsilon_{\mathrm{low}}, 1+\varepsilon_{\mathrm{high}}\Big)\hat{A}_{i,t}\right)\right] \\
\text{s.t.} \quad & 0 < \Big|\{o_i \mid \texttt{is\_equivalent}(a, o_i)\}\Big| < G.
\end{aligned}
$$

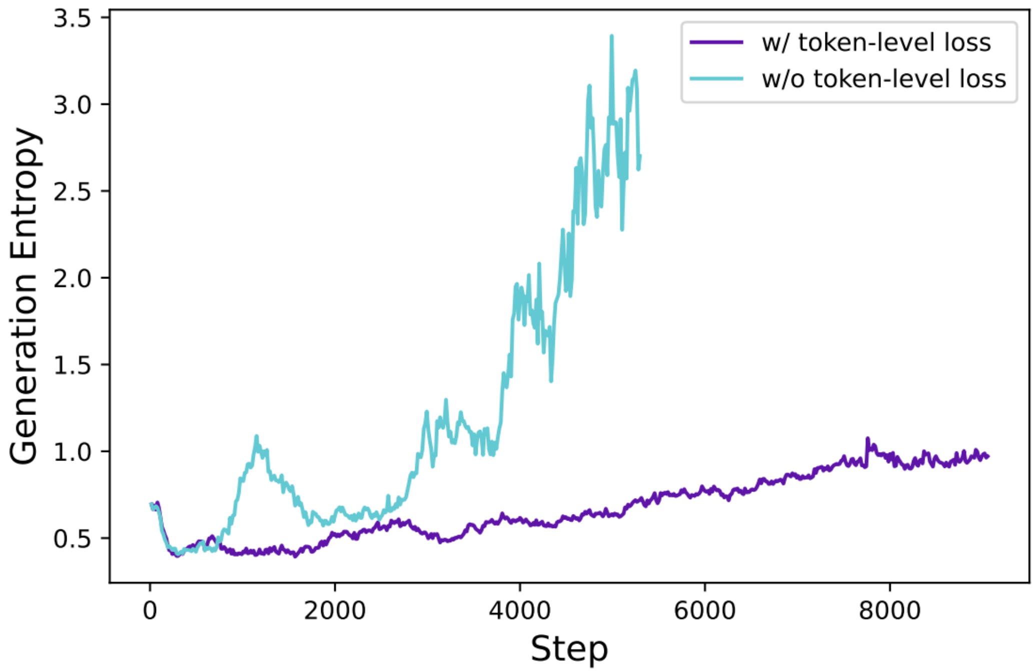
Token-Level Policy Gradient Loss

- The training is not stable and the model can collapse to output random words.
- Bad sequences typically occur under **a long context**.
  - Distribution of longer context is worse than short context.
- Sample-level loss underweights tokens of long sequences

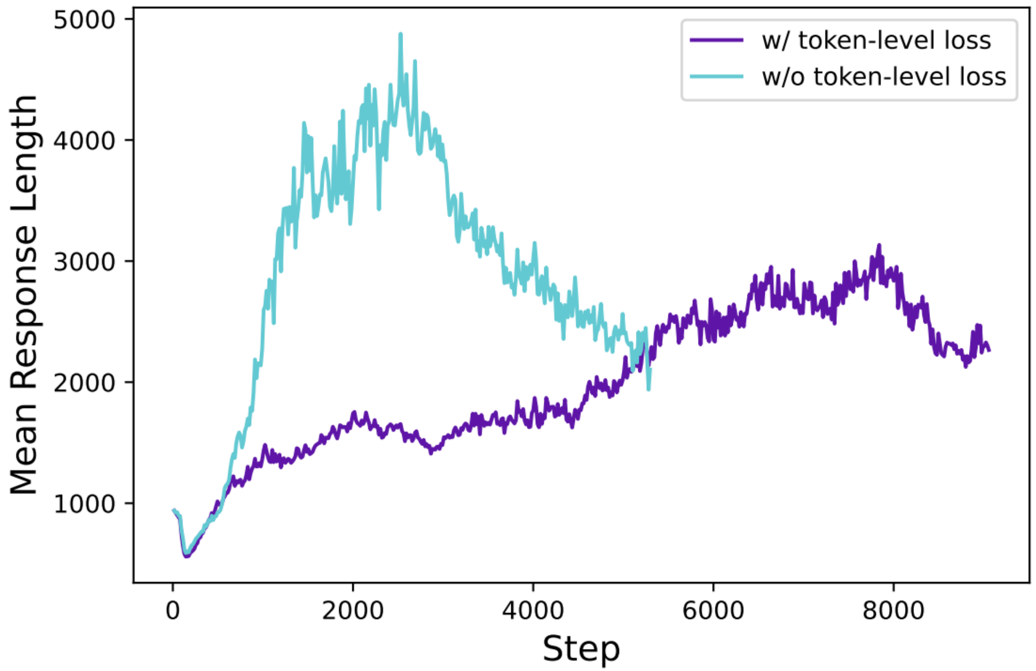
$$\mathcal{J}_{\mathrm{GRPO}}(\theta) = \mathbb{E}_{(q,a)\sim\mathcal{D},\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\mathrm{old}}}(\cdot|q)}$$

$$\left[\boxed{\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}}\left(\min\left(r_{i,t}(\theta)\hat{A}_{i,t},\ \mathrm{clip}\Big(r_{i,t}(\theta),1-\varepsilon,1+\varepsilon\Big)\hat{A}_{i,t}\right)-\beta D_{\mathrm{KL}}(\pi_\theta||\pi_{\mathrm{ref}})\right)\right],$$

Token-Level Policy Gradient Loss



(a) Entropy of actor model's generation probabilities.



(b) Average length of actor model-generated responses

Overlong Reward Shaping

- Overlong samples introduce **reward noise**
  - The typical max length is 16K, while the model can generate correct answers in 16K-24K.
- Method 1: mask overlong samples
  - However, 20% samples are masked, which is not suitable for scaling
- Method 2: soft overlong punishment
  - Decouple overlong punishment and correctness reward
  - Longer, more punishment

Performance

- Token-level loss brings marginal improvements while enhancing stability.

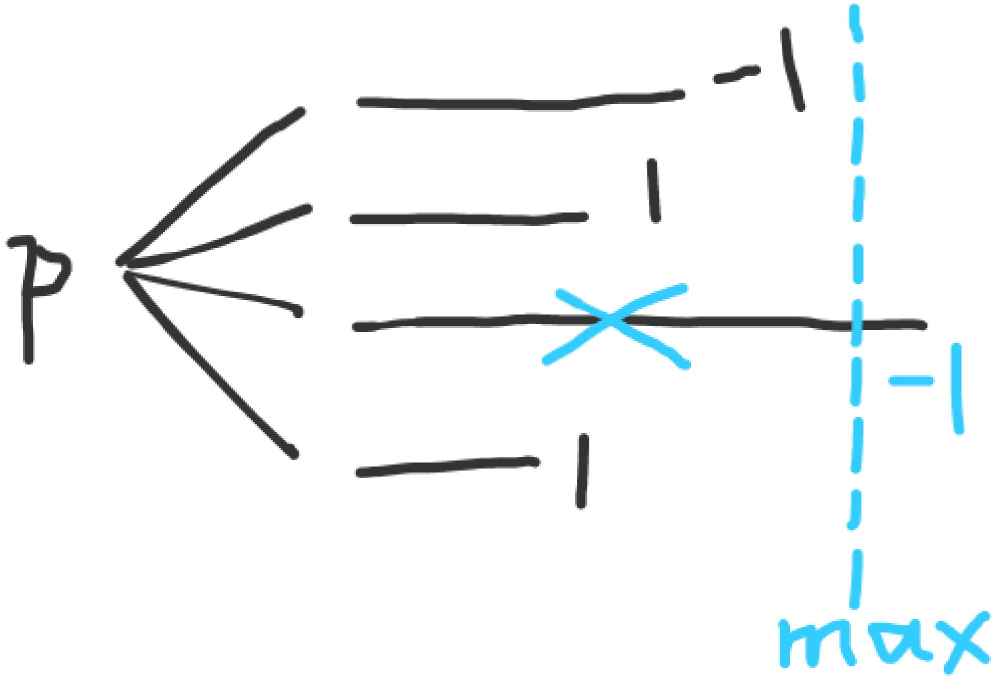
Table 1 Main results of progressive techniques applied to **DAPO**

Model	**AIME24**$_{\text{avg@32}}$
DeepSeek-R1-Zero-Qwen-32B	47
Naive GRPO	30
+ Overlong Filtering	36
+ Clip-Higher	38
+ Soft Overlong Punishment	41
+ Token-level Loss	42
+ Dynamic Sampling (**DAPO**)	**50**

VAPO

PPO v.s. GRPO

- The methods do not work in PPO
- GRPO just uses the group information
- Critic models can generalize across prompts



GAE

- Advantage Estimation
- λ controls the trade-off between the bias and variance.
- The default value is 0.95 for both policy and value.

- How to estimate the advantage:
  - Temporal-Difference (TD) estimation: $A_t^{(1)} := \delta_t^V = -V(s_t) + r_t + \gamma V(s_{t+1})$
  - Monte-Carlo (MC) estimation: $A_t^{(\infty)} := \sum_{l=0}^{\infty} \gamma^l \delta_{t+l}^V = -V(s_t) + \sum_{l=0}^{\infty} \gamma^l r_{t+l}$
  - GAE estimation：

$$A_t^{(1)} := \delta_t^V = -V(s_t) + r_t + \gamma V(s_{t+1})$$

$$A_t^{(2)} := \delta_t^V + \gamma \delta_{t+1}^V = -V(s_t) + r_t + \gamma r_{t+1} + \gamma^2 V(s_{t+2})$$

$$A_t^{(3)} := \delta_t^V + \gamma \delta_{t+1} + \gamma^2 \delta_{t+2}^V = -V(s_t) + r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \gamma^3 V(s_{t+3})$$

...

$$A_t^{(\infty)} := \sum_{l=0}^{\infty} \gamma^l \delta_{t+l}^V = -V(s_t) + \sum_{l=0}^{\infty} \gamma^l r_{t+l}$$

As λ increases, the bias decreases and the variance increases.

$$\begin{aligned}
A_t^{GAE(\gamma,\lambda)} :&= (1-\lambda)(A_t^{(1)} + \lambda A_t^{(2)} + \lambda^2 A_t^{(3)} + ...) \\
&= (1-\lambda)(\delta_t^V + \lambda(\delta_t^V + \gamma \delta_{t+1}^V) + \lambda^2(\delta_t^V + \gamma \delta_{t+1} + \gamma^2 \delta_{t+2}^V) + ...) \\
&= (1-\lambda)(\delta_t^V(1 + \lambda + \lambda^2 + ...) + \gamma \delta_{t+1}^V(\lambda + \lambda^2 + \lambda^3 ...) + ...) \\
&= (1-\lambda)(\delta_t^V \frac{1}{1-\lambda} + \gamma \delta_{t+1}^V \frac{\lambda}{1-\lambda} + ...) \\
&= \sum_{l=0}^{\infty} (\gamma\lambda)^l \delta_{t+l}^V
\end{aligned}$$

Enhanced Value Model Training

- Value-pretraining
  - To align with the initial policy and mitigate potential biases introduced by the value initialization.

- Decoupled-GAE:
  - Set the λ of the critic model to 1 instead of 0.95
  - The λ of the actor model remains 0.95 for faster convergence and reduced variance
  - To promote more unbiased value model training

Length-Adaptive GAE

- Backpropagation of the final reward decays exponentially, which disappears for longer sequences.

- To automatically handle sequences of varying lengths:
  we set the sum of λ to be proportional to **the output length**:

$$\sum_{t=0}^{\infty} \lambda_{\text{policy}}^{t} \approx \frac{1}{1-\lambda_{\text{policy}}} = \alpha l, \qquad \lambda_{\text{policy}} = 1 - \frac{1}{\alpha l}$$

Positive LM Loss

- To enhance the utilization of positive samples

$$\mathcal{L}_{\mathrm{NLL}}(\theta) = -\frac{1}{\sum_{o_i \in \mathcal{T}} |o_i|} \sum_{o_i \in \mathcal{T}} \sum_{t=1}^{|o_i|} \log \pi_\theta \left(a_t | s_t\right),$$

$$\mathcal{L}(\theta) = \mathcal{L}_{\mathrm{PPO}}(\theta) + \mu * \mathcal{L}_{\mathrm{NLL}}(\theta).$$

Thank you!