SAPO

SAPO

SAPO（Soft Adaptive Policy Optimization）论文总结

面对问题：在MoE模型中，**token级重要性比率**高方差会导致训练不稳定性。

解决方案：SAPO取代了GSPO和GRPO等现有方法中使用的**硬截断机制**，采用**平滑的、温度控制的门控函数**来连续地衰减偏离策略的更新，同时保留有用的学习信号。

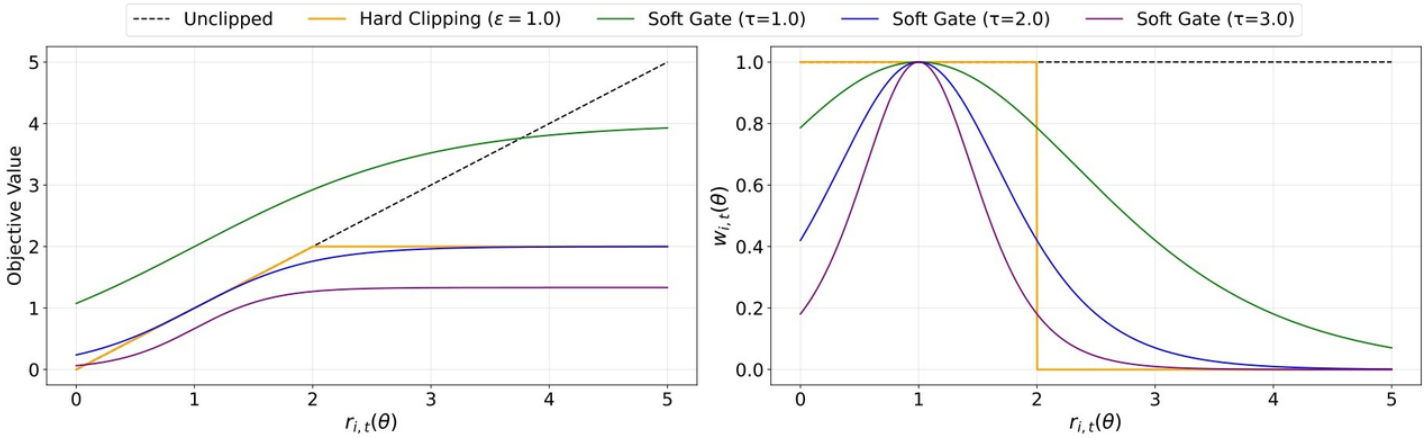
核心优势：兼具序列连贯性和token自适应性。

1. 保持了GSPO中的序列级别连贯性，通过**软门控**构建了一个连续的信任区域，避免了GSPO中硬截断。当序列中存在高度偏离策略的token时，SAPO能选择性地降低这些token的权重，而非像GSPO硬截断那样抑制整个序列的梯度，从而提升样本效率
2. SAPO还引入了**非对称温度设计**，通过对负向token梯度施加更快的衰减，显著提升了训练稳定性。

简介

- **当前主流的强化学习方法：group-based policy optimization**，即为每个查询采样多组响应，然后在组内对序列级别的奖励进行归一化，使用当前策略和采样策略之间的重要性比率来加权更新策略模型。
- **核心挑战：token级重要性比率（token-level importance ratios）的方差很高**，尤其在**MoE模型** 中，由于路由异构性和长响应，这种偏差会被放大，导致更新不稳定。
- **现有方法的局限性：GRPO**使用**硬截断（hard clipping）** 来约束大的偏差，即将固定范围之外($(1-\epsilon, 1+\epsilon)$ 之外)的梯度归零。这种方式虽然能抑制更新步长过大，但裁剪过紧会限制用于梯度计算的有效样本数量；裁剪过松则会引入来自off-policy样本的噪声梯度。 Qwen的前作 **GSPO** 也是通过硬截断的方式。
- **解决方案：软自适应策略优化（SAPO）**，用一个平滑的、温度控制的门控函数取代了硬截断，创新点如下：
  - **token-level软信任域**：如下左图所示，该软门控函数是重要性比率的一个**有界、S形（sigmoid）函数**，并以策略点（on-policy point）为中心。靠近在策略点的梯度得以保留，以鼓励有用的更新；当重要性比率偏离时，梯度平滑地衰减而不是截断，从而在保持一定学习信号的同时减少优化噪声。

- **非对称温度**：因为负token更新往往会增加许多不恰当的token的 logits，导致训练不稳定。为进一步提升大词表下的鲁棒性，SAPO 采用非对称温度来处理正向和负向token，即为负向token上梯度衰减更快。



- **序列连贯性与token自适应性**：
  - **对比 GSPO**：与 GSPO 相似，SAPO 保持了序列级别的连贯性，其软门控形成了一个连续的信任域。当序列中只有少数高度离策略的token时，GSPO 会抑制该序列的所有梯度，而 SAPO **选择性地仅降低**有问题的token的梯度权重，同时保留来自接近在策略点的token的学习信号，从而减轻了硬截断的信号损失并提高了样本效率。
  - **对比 GRPO**：相对于 GRPO，稳定性和任务性能都有所提升，能更长时间的连贯学习，在训练崩溃前实现更高的 Pass@1 性能。（所有算法训练到最后都会崩溃，但SAPO崩的时候效果最好。）

🏆 **回忆下GRPO和GSPO**

将参数为 θ 的自回归语言模型建模为一个对token序列的随机策略 π_θ，q 表示用户查询，D 表示查询集；y 表示对 q 的响应，$|y|$ 是 y 中token的数量，策略 π_θ 下响应 y 的似然为：$\pi_\theta(y|q) = \prod_{t=1}^{|y|} \pi_\theta(y_t|q, y_{<t})$

- **GRPO**：一种token-level的优化方法，对于每个查询 q，从行为策略 $\pi_{\theta_{\text{old}}}$ 中采样一组 G 个响应 $\{y_1, \ldots, y_G\}$，计算它们的奖励 $\{R_1, \ldots, R_G\}$，并最大化以下token-level目标函数：

$$\mathcal{J}_{\text{GRPO}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\{y_i\}_{i=1}^{G}\sim\pi_{\text{old}}(\cdot|q)} \left[\frac{1}{G}\sum_{i=1}^{G} \frac{1}{|y_i|} \sum_{t=1}^{|y_i|} \min\left(r_{i,t}(\theta)\hat{A}_{i,t}, \text{clip}\left(r_{i,t}(\theta), 1-\varepsilon, 1+\varepsilon\right)\hat{A}_{i,t} \right) \right], \quad (1)$$

其中，重要性比率 $r_{i,t}(\theta)$ 和群组归一化优势 $\hat{A}_{i,t}$ 定义为：

$$r_{i,t}(\theta) = \frac{\pi_\theta(y_{i,t}|q, y_{i,<t})}{\pi_{\theta_{\text{old}}}(y_{i,t}|q, y_{i,<t})}, \quad \hat{A}_{i,t} = \hat{A}_i = \frac{R_i - \text{mean}\left(\{R_j\}_{j=1}^{G}\right)}{\text{std}\left(\{R_j\}_{j=1}^{G}\right)}, \quad (2)$$

$\epsilon > 0$ 表示clip范围，G 是组中响应的数量，同一响应内的所有token中共享优势 $\hat{A}_i$ 。

- **GSPO**：在**sequence-level（序列级别）**应用clip，保持奖励和重要性比率同一粒度，通过在 $s_i(\theta)$ 进行长度归一化降低方差。采用以下**sequence-level**的优化目标：

$$\mathcal{J}_{\text{GSPO}}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\{y_i\}_{i=1}^{G}\sim\pi_{\text{old}}(\cdot|q)}\left[\frac{1}{G}\sum_{i=1}^{G}\min\left(s_i(\theta)\hat{A}_i, \text{clip}(s_i(\theta), 1-\epsilon, 1+\epsilon)\hat{A}_i\right)\right], \quad (3)$$

$$s_i(\theta) = \left(\frac{\pi_\theta(y_i|q)}{\pi_{\text{old}}(y_i|q)}\right)^{\frac{1}{|y_i|}} = \exp\left(\frac{1}{|y_i|}\sum_{t=1}^{|y_i|}\log\frac{\pi_\theta(y_{i,t}|q,y_{i,<<t})}{\pi_{\text{old}}(y_{i,t}|q,y_{i,<<t})}\right), \quad \hat{A}_i = \frac{R_i - \text{mean}\left(\{R_j\}_{j=1}^{G}\right)}{\text{std}\left(\{R_j\}_{j=1}^{G}\right)} \quad (4)$$

GRPO和GSPO都是用了硬截断，即 $\text{clip}\left(r_{i,t}(\theta), 1-\varepsilon, 1+\varepsilon\right)$ ，将超过 $(1-\epsilon, 1+\epsilon)$ 范围外的重要性比率都会直接截断，对应上左图的黄色线。

SAPO算法

SAPO训练最大化以下目标：

$$\mathcal{J}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\{y_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|q)}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|y_i|}\sum_{t=1}^{|y_i|}f_{i,t}\left(r_{i,t}(\theta)\right)\hat{A}_{i,t}\right], \quad (5)$$

$$f_{i,t}(x) = \sigma\left(\tau_{i,t}\left(x-1\right)\right)\cdot\frac{4}{\tau_{i,t}}, \quad \tau_{i,t} = \begin{cases}\tau_{\text{pos}}, & \text{if } \hat{A}_{i,t} > 0\\ \tau_{\text{neg}}, & \text{otherwise}\end{cases}, \quad (6)$$

其中重要性比率 $r_{i,t}(\theta)$ 和优势 $\hat{A}_{i,t}$ 的计算方式与 GRPO 相同（公式 (2)），$\sigma(x) = 1/(1+e^{-x})$ 是 Sigmoid 函数，τ_{pos} 和 τ_{neg} 分别是对应于**正优势**和**负优势**token的温度参数。下面结合SAPO的原理看下是如何实现SAPO的两个创新点的

token-level软信任域

- 之所以说SAPO是token-level的，是因为**SAPO是在token粒度进行梯度计算**。当策略步长小且序列中没有off-policy token 时，其平均token门控会集中为一个平滑的序列级门控 $g\left(\log s_i(\theta)\right) = \text{sech}^2\left(\frac{G}{2}\log s_i(\theta)\right)$ ，类似于GSPO中的形式。当序列中出现少数off-policy token时，SAPO只会抑制这些token的梯度，提高了样本效率，而GSPO会抑制整个序列的梯度。
- 软信任域对应目标函数中的 $f_{i,t}(x)$ 项，这是一个有界的sigmoid形状的函数，用于对重要性比率进行调整。对目标函数进行求导得到加权对数策略梯度：

$$\nabla_\theta\mathcal{J}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\{y_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|q)}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|y_i|}\sum_{t=1}^{|y_i|}w_{i,t}(\theta)r_{i,t}(\theta)\nabla_\theta\log\pi_\theta(y_{i,t}\mid q,y_{i,<<t})\right], \quad (7)$$

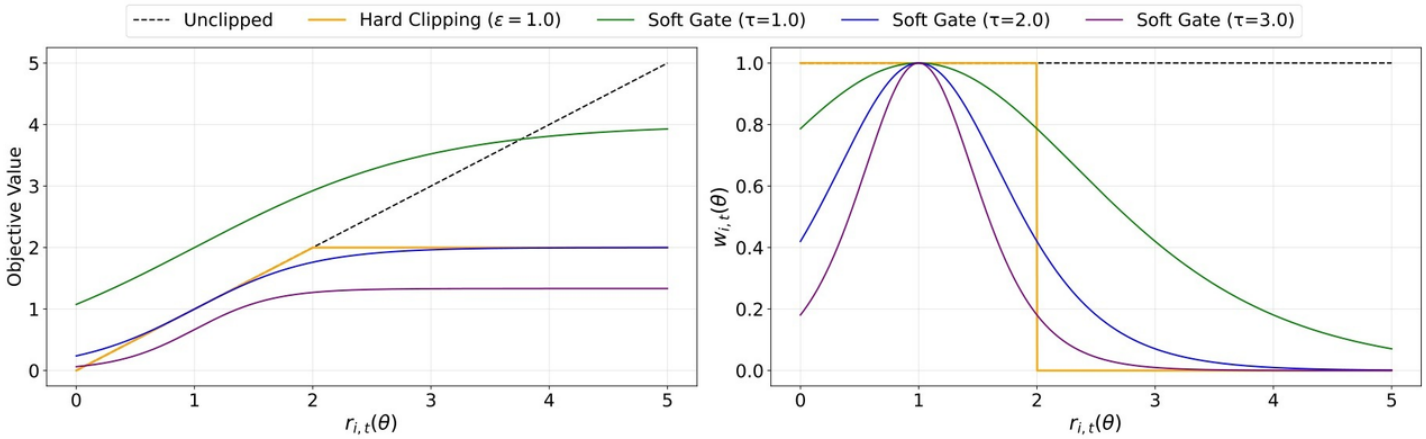
$$w_{i,t}(\theta) = 4p_{i,t}(\theta)\left(1-p_{i,t}(\theta)\right), \quad p_{i,t}(\theta) = \sigma\left(\tau_{i,t}\left(r_{i,t}(\theta)-1\right)\right), \quad (8)$$

- 当 $r_{i,t}(\theta)=1$ 时，策略梯度达到最大值，$w_{i,t}(\theta)=1$（如上右图），软门控梯度等于未截断目标函数 $r_{i,t}(\theta)A_{i,t}$ 的梯度，与 $\tau_{i,t}$ 无关。当重要性比率 $r_{i,t}(\theta)$ 偏离1时，梯度会被**平滑地近似指数的衰减**，从而实现了**连续软信任域**，如上左图蓝、绿、紫线。而不是像GSPO/GRPO中硬截断那样被截断为零，这样能防止更新过大，也能防止梯度消失，保留了一定的学习信号。

非对称温度

SAPO对优势为正和负的样本采用不同的温度：

$$\tau_{i,t}=\begin{cases}\tau_{\text{pos}}, & \text{if } \hat{A}_{i,t}>0\\ \tau_{\text{neg}}, & \text{otherwise}\end{cases}$$



> 左图表示优势为正时目标函数与重要性比率的关系；右图表示梯度权重 $w_{i,t}(\theta)$ 与重要性比率的关系.

温度决定了 $w_{i,t}(\theta)$ 随 $r_{i,t}(\theta)$ 变化的曲线。如上图右所示，随着重要性比率 $r_{i,t}(\theta)$ 逐步偏离1，温度 τ 越大，梯度的权重 $w_{i,t}(\theta)$ 衰减的越快。那为什么要设置 $\tau_{neg}>\tau_{pos}$ 呢?

已知负向token 会增加异常的token的 logits，更容易引入不稳定性。设 $z=[z_1,z_2,\ldots,z_{|V|}]$ 表示 logits，v 表示一个token，通过softmax计算输出概率，$\pi_\theta(v\mid q,y_{i,<t})=\exp(z_v)/\sum_{v'}\exp(z_{v'})$，策略梯度的更新为：

$$\begin{aligned}\frac{\partial \log \pi_\theta(y_{i,t}\mid q,y_{i,<<t})\hat{A}_{i,t}}{\partial z_v}&=\frac{\partial \pi_\theta(y_{i,t}\mid q,y_{i,<<t})}{\partial z_v}\cdot\frac{\hat{A}_{i,t}}{\pi_\theta(y_{i,t}\mid q,y_{i,<<t})}\\&=\frac{\mathbb{I}(v=y_{i,t})\exp(z_{y_{i,t}})\sum_{v'\in\mathcal{V}}\exp(z_{v'})-\exp(z_{y_{i,t}})\exp(z_v)}{\left(\sum_{v'\in\mathcal{V}}\exp(z_{v'})\right)^2}\cdot\frac{\hat{A}_{i,t}}{\pi_\theta(y_{i,t}\mid q,y_{i,<<t})}\\&=\begin{cases}(1-\pi_\theta(y_{i,t}\mid q,y_{i,<<t}))\cdot\hat{A}_{i,t} & \text{if } v=y_{i,t}\quad\text{(sampled token)}\\ -\pi_\theta(v\mid q,y_{i,<<t})\cdot\hat{A}_{i,t} & \text{otherwise}\quad\text{(unsampled token)}\end{cases}\end{aligned}$$

🏆 **公式解释**

公式第一行是因为求导链式法则 $\frac{\partial \log \pi_\theta(y_{i,t}|\cdot)\cdot \hat{A}_{i,t}}{\partial z_v} = \underbrace{\frac{\partial \pi_\theta(y_{i,t}\mid\cdot)}{\partial z_v}}_{\text{softmax梯度}} \cdot \underbrace{\frac{\hat{A}_{i,t}}{\pi_\theta(y_{i,t}\mid\cdot)}}_{\text{对数梯度的链式法则}}$

公式第二行已知 $\pi_\theta\left(v \mid q, y_{i,<t}\right) = \exp\left(z_v\right) / \sum_{v'} \exp\left(z_{v'}\right)$ 是softmax函数，求导为：

- 当 $v = y_{i,t}$ （当前采样的 token）：$\frac{\partial \pi_\theta(y_{i,t}|\cdot)}{\partial z_v} = \pi_\theta(y_{i,t}\mid\cdot)\cdot(1-\pi_\theta(y_{i,t}\mid\cdot))$
- 当 $v \neq y_{i,t}$ （其他 token）：$\frac{\partial \pi_\theta(y_{i,t}|\cdot)}{\partial z_v} = -\pi_\theta(y_{i,t}\mid\cdot)\cdot\pi_\theta(v\mid\cdot)$

公式第二行通过指示函数 $\mathbb{I}(v = y_{i,t})$ 把两种情况合并，分子是“$v = y_{i,t}$ 时的梯度分子”，分母是 softmax 归一化项的平方。

可以看到正向优势会提高采样token v 的对数概率，同时降低所有未采样token的对数概率；而负向优势则起到相反作用，提升许多未采样token的对数概率。由于LLM 词表巨大，**负优势会扩散到许多不相关的token从而导致不稳定性**。因此SAPO 采用**非对称温度**（$\tau_{neg} > \tau_{pos}$）来增强训练的鲁棒性，温度 τ 越大时这样让负向token的梯度衰减得比正向token更快，显著降低了训练早期崩溃的可能性，提高了稳定性和性能。

重要性比率 $r_{i,t}(\theta)$ 与优势的关系

$r_{i,t}(\theta)$ 实际反映的是当前策略与采样策略之间的差异，策略差异越大时（$r_{i,t}(\theta)$ 越偏离1）这个token的梯度权重就越小，这个token对策略更新的影响也就越小。而负优势会导致不相关的token的概率提升。两者相结合会有以下效果：

- **优势为负且策略差异大**：可能会导致不相关token的概率大幅提升，从而引发训练不稳定，SAPO中选择降低其梯度权重从而降低其负面作用。
- **优势为负且策略差异小**：会提供一定的正则化效果，扩大了采样空间和防止过拟合，应当保留。
- **优势为正**：鼓励有益的更新和探索，应当保留

从门控函数视角统一SAPO/GSPO/GRPO

引入SAPO/GSPO/GRPO的统一目标函数：

$$\mathcal{J}(\theta) = \mathbb{E}_{q\sim\mathcal{D},\{y_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(\cdot|q)}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|y_i|}\sum_{t=1}^{|y_i|} f_{i,t}(r_{i,t}(\theta))\hat{A}_{i,t}\right], (10)$$

其中 $f_{i,t}(\cdot)$ 是特定算法的**门控函数**。GSPO中的长度归一化的序列级比率 $s_i(\theta)$ 为token比率的几何平均值：

$$s_i(\theta) = \left(\frac{\pi_\theta(y_i|q)}{\pi_{\theta_{\text{old}}}(y_i|q)} \right)^{\frac{1}{|y_i|}} = \exp\left(\frac{1}{|y_i|} \sum_{t=1}^{|y_i|} \log r_{i,t}(\theta) \right), \quad s_{i,t}(\theta) = \text{sg}\left[s_i(\theta) \right] \cdot \frac{\pi_\theta(y_{i,t}|q, y_{i,<<t})}{\text{sg}\left[\pi_\theta(y_{i,t}|q, y_{i,<<t}) \right]}, (11)$$

$sg[\cdot]$ 表示取数值但停止梯度传播（即Pytorch 中的detach）。不同算法的门控函数 $f_{i,t}(\theta)$ 为：

- SAPO：采用token-level的、平滑的函数作为门控，采用非对称温度。当策略变化变大时，梯度平滑地衰减而不是直接截断。

$$f_{i,t}^{\text{SAPO}}(r_{i,t}(\theta)) = \frac{4}{\tau} \sigma\left(\tau_i (r_{i,t}(\theta) - 1) \right), \quad \tau_i = \begin{cases} \tau_{\text{pos}}, & \hat{A}_i > 0, \\ \tau_{\text{neg}}, & \hat{A}_i \le 0, \end{cases} (12)$$

- GRPO：采用token-level的、硬截断作为门控函数。信任域 $(1-\epsilon, 1+\epsilon)$ 内token的梯度与未截断一致，信任域外的token没有梯度，容易导致训练不稳定。

$$f_{i,t}^{\text{GRPO}}(r_{i,t}(\theta); \hat{A}_i) = \begin{cases} \min(r_{i,t}(\theta), 1+\varepsilon), & \hat{A}_i > 0, \\ \max(r_{i,t}(\theta), 1-\varepsilon), & \hat{A}_i \le 0, \end{cases} (13)$$

- GSPO：采用sequence-level的、硬截断作为门控函数。当序列中有高度偏离策略token时，会抑制整个序列的梯度。

$$f_{i,t}^{\text{GSPO}}(r_{i,t}(\theta); \hat{A}_i) \equiv f_{i,t}^{\text{seq}}(s_{i,t}(\theta); \hat{A}_i) = \begin{cases} \min(s_{i,t}(\theta), 1+\varepsilon), & \hat{A}_i > 0, \\ \max(s_{i,t}(\theta), 1-\varepsilon), & \hat{A}_i \le 0. \end{cases} (14)$$

✏️ **总结下论文中推导的结论**

- **SAPO相比于GRPO**：都是**token-level的门控函数**，通过软门控缓解了硬截断带来的梯度消失，增强训练的稳定性。采用**非对称温度设计**，让负token的梯度更快衰减，避免负优势扩散到不相关token。
- **SAPO相比于GSPO**：同样通过软门控缓解了硬截断带来的梯度消失，增强训练的稳定性。并且条件温和的情况下，SAPO可以简化为类似GSPO的序列级方法；条件不温和情况下，SAPO能简化类似于为GRPO token-level方法，只对高度偏离策略的token进行抑制，而不是像SAPO中抑制整个序列的梯度。

至于怎么判断更新条件是否温和，论文提出了两个假设，当满足假设时则视为条件温和。

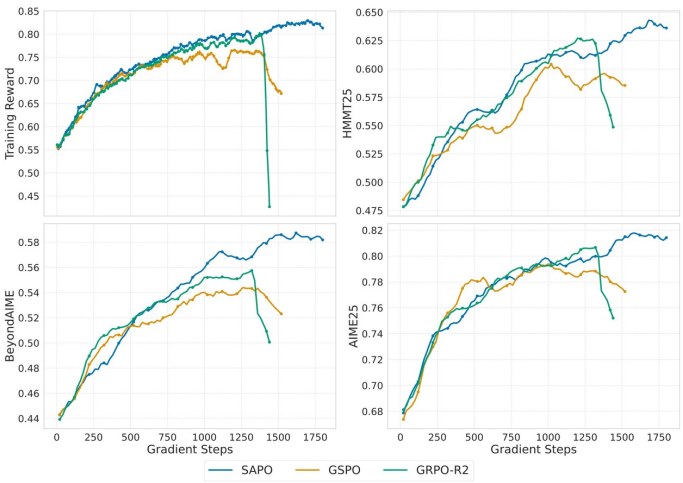
1. **步长小且on-policy**：假设新旧策略接近 $r_{i,t}(\theta) \approx 1$，此时可以近似于一阶泰勒展开 $\log r_{i,t}(\theta) \approx r_{i,t}(\theta) - 1$。
2. **序列内分散度低**：即序列中大部分token的更新幅度接近。令 $z_{i,t}(\theta) := \log r_{i,t}(\theta)$，$\mu_i(\theta) := \frac{1}{|y_i|} \sum_t z_{i,t}(\theta) = \log s_i(\theta)$ 表示每个token更新频率的均值。对于大多数序列，方差 $\text{Var}_i(\theta) := \frac{1}{|y_i|} \sum_t (z_{i,t}(\theta) - \mu_i(\theta))^2$ 较小。

具体的推导感兴趣去论文中自行研究，这里不做详细展开。

实验

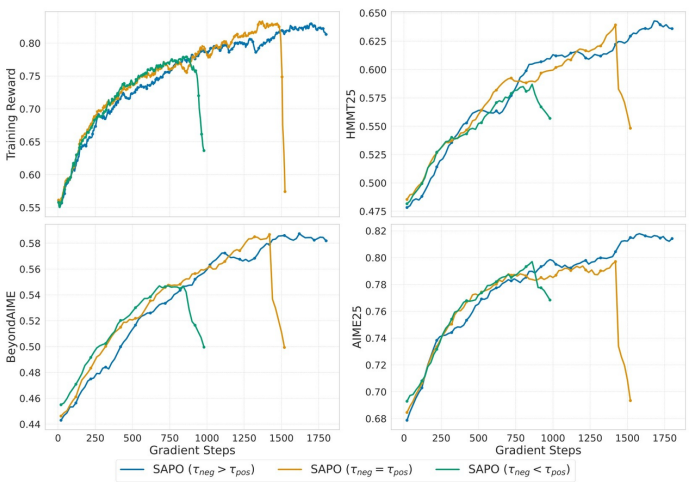
推理任务

- **模型：** **Qwen3-30B-A3B-Base 的冷启动模型**
- **Benchmark：** 训练奖励，以及在三个基准测试（AIME25、HMMT25 和 BeyondAIME）上的16个样本的平均 Pass@1 验证性能。
- **对比算法：** 将 SAPO 与 GSPO 和 GRPO-R2（路由回放的 GRPO）进行比较。
- **超参数设置：** $\tau_{pos} = 1.0$、$\tau_{neg} = 1.05$ ，每个rollout数据分成4个mini-batch进行梯度更新。
- **实验结果：**SAPO 在所有基准测试中持续提高了模型性能，而GSPO 和 GRPO-R2 均在训练初期出现崩溃，SAPO实现了更高的稳定性和更强的最终性能。
- **无需路由回放：**SAPO **不需要依赖路由回放**（routing replay，GRPO-R2 使用的技术）有助于提高探索效率并减少 RL 系统的工程开销。



温度消融实验

- **配置对比：** 评估了三种配置：
  - $\tau_{neg} = 1.05 > \tau_{pos} = 1.0$ （非对称，负向更高）
  - $\tau_{neg} = \tau_{pos} = 1.0$ （对称）
  - $\tau_{neg} = 0.95 < \tau_{pos} = 1.0$ （非对称，负向更低）。
- **实验结果：** 当负向token分配**更高温度** 时，训练**最稳定**。而当负向token分配**更低温度**时，训练**最不稳定**。证明负token的梯度更容易导致训练不稳定。



Qwen3-VL训练

- **模型与任务：** SAPO 应用于 Qwen3-VL 模型系列的训练，涉及各种尺寸的模型（包含 MoE 和 Dense架构）。训练任务包括文本和多模态任务的广泛集合，如数学、编码和逻辑推理。
- **训练设置：** 采用多任务学习，在每个批次内保持固定的任务采样比例。使用大 batch-size，并将每个批次的 rollout 数据分成两个 mini-batch 进行梯度更新。
- **实验结果：** SAPO 在整个训练过程中取得了稳定的性能提升，在相同的计算预算下，训练奖励和验证集上的指标都更优