GFPO

GFPO

论文总结

尽管增大响应长度，可以提升在复杂问题上的准确率，但却会让大模型变成话痨，即**许多token是作为填充内容，对推理没有实际意义**。GFPO通过对每个问题采样更大的响应组，并基于回复长度和token效率（奖励与响应长度的比值）过滤回复，教会模型推理时减少思考（**训练时更多采样，推理时更少思考**）。还可以根据问题难度动态调整分配的计算资源，实现在复杂问题上计算效率和准确率的平衡。

GRPO

$$\widehat{A}_{i,t} = \frac{R(q,o_i) - \text{mean}\{R(q,o_1),\ldots,R(q,o_G)\}}{\text{std}\{R(q,o_1),\ldots,R(q,o_G)\}}$$

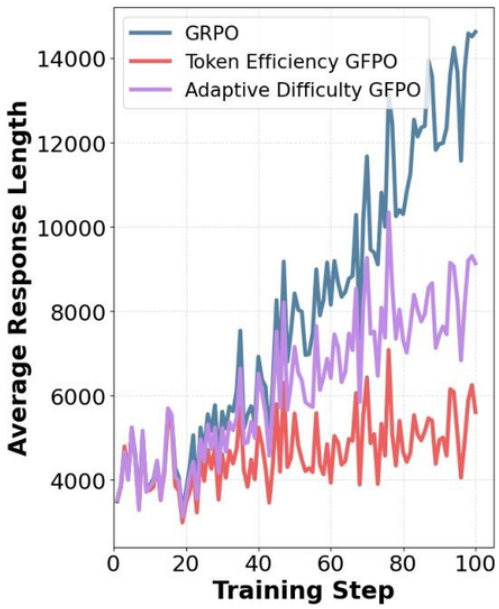
$$\mathcal{L}_{GRPO_{i,t}} = \min\left[\frac{\pi_\theta(o_{i,t} \mid q,o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q,o_{i,<t})}\widehat{A}_{i,t}, \text{clip}\left(\frac{\pi_\theta(o_{i,t} \mid q,o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t} \mid q,o_{i,<t})}, 1-\varepsilon, 1+\varepsilon\right)\widehat{A}_{i,t}\right]$$

GFPO

$$\mathcal{S} = \text{RejectionSample}(\mathcal{G}, k, \text{metric}),\ m_i = \mathbb{I}_{\{i\in\mathcal{S}\}}$$

$$\widehat{A}^{(m)}_{i,t} = \frac{R(q,o_i) - \text{mean}\{R(q,o_{s_1}),\ldots,R(q,o_{s_k})\}}{\text{std}\{R(q,o_{s_1}),\ldots,R(q,o_{s_k})\}} m_i$$

$$\mathcal{L}_{GFPO_{i,t}} = \min\left[\frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}|q,o_{i,<t})}\widehat{A}^{(m)}_{i,t}, \text{clip}\left(\frac{\pi_\theta(o_{i,t}|q,o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}|q,o_{i,<t})}, 1-\varepsilon, 1+\varepsilon\right)\widehat{A}^{(m)}_{i,t}\right]$$



参考论文：《Sample More to Think Less: Group Filtered Policy Optimization for Concise Reasoning》

- **大模型为什么会变成话唠?**

基于奖励验证的强化学习算法（GRPO、PPO等），倾向于给答案正确且推理步骤更长的回答更高的奖励，随着训练的进行，模型就倾向于用更长的回复来换取更高的奖励，即便回复中的大多数token对推理没有贡献。

- **更长的推理链不会带来更优的回复**

更长的推理链不等同于更好的推理能力，因为很多token对推理没有用处，而只是作为填充（例如反复自我验证、事无巨细的解释），特别是在比较难的问题上。在AIME 25上作者发现，当同时生成了正确和错误的回答时，在72%的题目中，较长的回答比其较短的对应回答更有可能是错误的。

- **论文创新：**

- **GFPO(Group Filtered Policy Optimization）**：将**拒绝采样**与GRPO结合，每个问题采样更大的推理链组 G 并保留top-k的推理链，以增加理想输出出现的概率。拒绝采样重点关注**响应长度**，要保留更短的推理链。
- **Token 效率**：定义为奖励与响应长度的比值，反映推理链长度的合理性。
- **自适应难度GFPO**：根据问题难度，使用无监督的问题难度估计，动态调整保留响应数量 k ，为更难的问题分配更多计算资源，即生成更多样的回复。

- **GFPO的本质是训练与推理阶段计算资源的平衡**

增加训练时间的计算资源可直接转化为测试时间计算资源的减少，这一转化是极为有利的，因为训练时的资源投入是一次性的，推理的成本节约则可以在部署后持续受益。GFPO通过在训练阶段引入过滤机制，能在不牺牲推理能力的情况下，让模型生成更简洁的推理过程。

✏️ **回顾下GRPO**

GRPO为每个问题采样一组多个响应，并使用组内平均奖励计算优势值，从而省去了评估模型。令 θ 表示模型参数， q 表示问题， o 表示从旧策略 $\pi_{\theta_{old}}$ 采样的响应， $R(q,o_i)$ 表示回复的奖励，优势为 $\hat{A}_{i,t}=\frac{R(q,o_i)-\frac{1}{k}\sum_{j=1}^{G}R(q,o_j)}{\sqrt{\frac{1}{k}\sum_{i=1}^{G}\left(R(q,o_i)-\frac{1}{k}\sum_{j=1}^{G}R(q,o_j)\right)^2}}$ ，重要性比率为 $r_{i,t}=\frac{\pi_\theta(o_{i,t}\mid q,o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}\mid q,o_{i,<t})}$ ，KL-惩罚为 $\beta\,\mathcal{D}_{KL}(\pi_\theta\,\|\,\pi_{\theta_{\text{old}}})$ ， $Entropy(\cdot)$ 是熵正则项，鼓励探索：

$$\mathcal{J}_{\text{GRPO}}(\theta)=\mathbb{E}_{[q\sim P(Q),\{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(O|q)]}\left[\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}\min\left(r_{i,t}\hat{A}_{i,t},\ \text{clip}(r_{i,t},1-\epsilon,1+\epsilon)\hat{A}_{i,t}\right)\right.$$
$$\left.-\beta\,\mathcal{D}_{\text{KL}}(\pi_\theta\,\|\,\pi_{\theta_{\text{old}}})+\gamma\,\text{Entropy}(\pi_\theta)\right]$$

注意，本文在实现GRPO时，中将上式中的 $\frac{1}{G}\sum_{i=1}^{G}\frac{1}{|o_i|}\sum_{t=1}^{|o_i|}$ 要替换成DAPO 的 token 级损失归一化 $\frac{1}{\sum_{i=1}^{G}|o_i|}\sum_{i=1}^{G}\sum_{t=1}^{|o_i|}$ 。

GRPO依赖单一的奖励信号，难以同时优化多个目标，如准确性和简洁性，往往会为了准确性而大幅增加响应长度，而GFPO可以实现对中各目标同时进行优化。

GFPO定义

GFPO的思路：**为每个响应采样更大一组的候选响应，以增加理想输出出现的概率，在计算优势和梯度之前，对采样出的回答再进行过滤。**

GFPO可以分为几个步骤：

1. **拒绝采样**：给定问题 q ，从旧策略采样一大组的响应 G 。然后根据指定的指标（比如**响应长度，token效率**），筛选出得分最高的k个响应，构成子集 S 。一定二维掩码 $m \in \{0,1\}^G$ ， m_i 表示选择的响应。
2. **策略更新**：在子集 S 中计算梯度，其余被过滤掉的回复不参与参数更新。这是一种隐式的更灵活的奖励构造方式，不需要设计复杂的奖励函数，就可以同时实现多个目标，GFPO的优化目标为：

$$\mathcal{J}_{\text{GFPO}}(\theta) = \mathbb{E}_{q\sim P(Q),\, \{o_i\}_{i=1}^{G}\sim\pi_{\theta_{\text{old}}}(O|q)} \frac{1}{\sum_{i=1}^{G}|o_i|} \sum_{i=1}^{G}\sum_{t=1}^{|o_i|} \min\Big(r_{i,t}\,\widehat{A}_{i,t}^{(m)}, \text{clip}(r_{i,t}, 1-\varepsilon, 1+\varepsilon)\,\widehat{A}_{i,t}^{(m)}\Big)$$

$$-\beta\,\mathcal{D}_{KL}\big(\pi_\theta \,\|\, \pi_{\theta_{\text{old}}}\big) + \gamma\,\text{Entropy}(\pi_\theta) \tag{2}$$

$$\mu_S = \frac{1}{k}\sum_{i\in S} R(q,o_i), \qquad \sigma_S = \sqrt{\frac{1}{k}\sum_{i\in S}\big(R(q,o_i)-\mu_S\big)^2}, \qquad \widehat{A}_{i,t}^{(m)} = \frac{R(q,o_i)-\mu_S}{\sigma_S} m_i.$$

$$r_{i,t} = \frac{\pi_\theta(o_{i,t}\,|\,q,o_{i,<t})}{\pi_{\theta_{\text{old}}}(o_{i,t}\,|\,q,o_{i,<t})}.$$

$$S, m = REJECTIONSAMPLE(G, k, \text{metric}, \text{order}), \quad m_i = \mathbb{I}_{\{i\in S\}}$$

其中不在 S 中的响应优势值为0，GFPO虽然采样更多响应，但训练目标中包含生成更短的响应，不会导致计算成本大幅增加，本文的评分指标为：

1. **响应长度**: 短响应上训练可以直接鼓励模型生成更简洁的输出。
2. **token效率：**定义为奖励与响应长度的比值，鼓励模型生成准确且简洁的响应。
3. 接下来介绍的自适应难度GFPO。

自适应难度GFPO

GFPO虽然鼓励生成更简短的推理链，但难题可能需要更长的推理链才能保障回复的准确率，自适应难度的GFPO就是要根据问题难度，动态调整过滤掉粒度。

- **难度衡量标准：**用每个问题采样响应的平均奖励来估计问题难度，平均奖励低表示问题更难。
- **自适应调整**：使用类似t-digest数据结构，维护一个问题难度的流式数据摘要，计算历史问题难度的四分位数（25%，50%，75%），将当前问题归类到对应难度的桶中，简单问题保留4个响应，中等问题保留6个响应，困难和极难问题保留8个响应。在初始阶段的预热训练中，所有问题都保留8个回复，以防止因问题不足导致的难度估计偏差。难度划分数量和每个难度保留的响应数量 k 是可调整的超参数。

实验分析

- **基座模型**：Phi-4-reasoning
- **对照实验**：对比了GRPO和GFPO算法，GRPO的组大小为8，GFPO测试了 $G \in \{8,16,24\}, k \le 8$ 。
- **数据集**：7.2万数学题，RL训练100步，batch-size为64。

- **奖励函数：**设置为二值准确性奖励 R_{acc} 与n-gram重复奖励的加权和， $LENGTHSCALE$ 表示对正确的常会都进行惩罚，这个奖励函数自身的长度惩罚不足以抑制GRPO的长度膨胀。

$$R = w_{\text{acc}}\text{LengthScale}(R_{\text{acc}}) + w_{\text{rep}}R_{\text{rep}}, \tag{3}$$

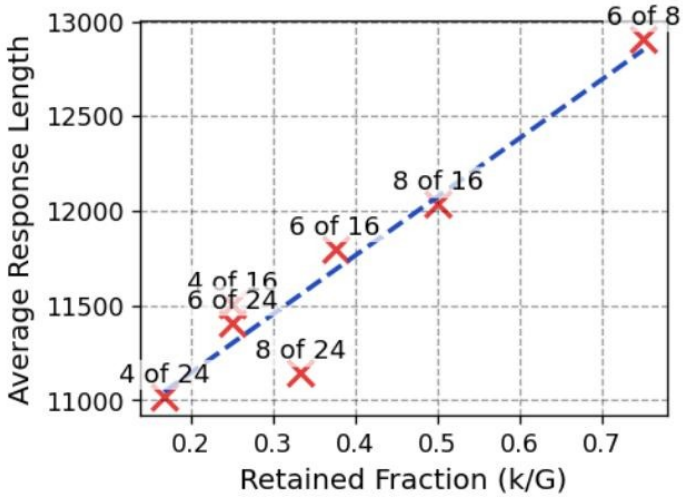
- **评测指标：**pass@1、原始响应长度 L，超额长度削减 $ELR = \frac{(L_{GRPO} - L_{GFPO})}{(L_{GRPO} - L_{SFT})}$ ，衡量GFPO相较于 SFT 模型，降低 GRPO 引起的响应长度膨胀的程度。

	AIME 25			AIME 24			GPQA		
	Acc	Avg Len	% Len Inf (↓)	Acc	Avg Len	% Len Inf (↓)	Acc	Avg Len	% Len Inf (↓)
SFT	64.2	10.9k	N/A	72.2	10.1k	N/A	67.0	6.6k	N/A
GRPO	72.4	14.8k	0.0	77.7	13.3k	0.0	67.5	10.7k	0.0
6 of 8	69.2	14.7k	1.8	79.6	13k	9.5	70.2	10.2k	11.5
8 of 16	**70.2**	13.9k	23.8	**77.9**	12.3k	33.0	**70.0**	9.7k	23.7
6 of 16	70.1	13.8k	25.6	76.9	12.2k	35.6	68.3	9.1k	38.8
4 of 16	69.7	**13.3k**	**38.0**	76.6	**11.8k**	**46.8**	68.6	**8.8k**	**45.7**
8 of 24	**70.4**	**12.6k**	**54.4**	75.1	11.6k	52.7	68.9	8.6k	52.2
6 of 24	68.5	13.1k	41.0	75.6	11.9k	44.9	**70.2**	8.7k	48.6
4 of 24	70.3	13k	46.1	**76.5**	**11.3k**	**59.8**	68.1	**8.3k**	**57.3**
Token Efficiency	69.5	**12k**	**70.9**	76.4	**10.6k**	**84.6**	68.5	**7.5k**	**79.7**
Adaptive Difficulty	**70.8**	12.8k	50.8	**76.6**	11.6k	52.9	**70.8**	9k	41.7

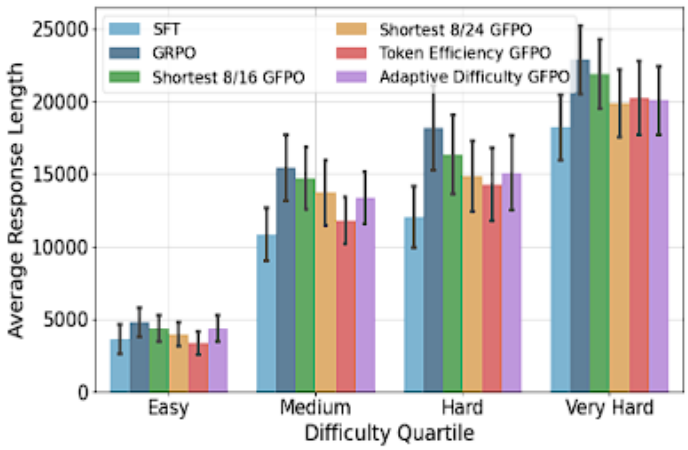
> 上图中6 of 8 、8 of 16等表示按照从G中保留k个最短的回复，Token Efficiency 表示保留k个token效率最高的响应，Adaptive Difficulty表示自适应GFPO

实验结果如上图，可以观察到GFPO能有效降低token使用量（平均长度大幅下降和ELR普遍大于0），并且能取得与GRPO相匹配的推理能力。接下来是这篇论文的最有价值的部分，通过大量实验分析了GFPO和GRPO在不同设置下的特点：

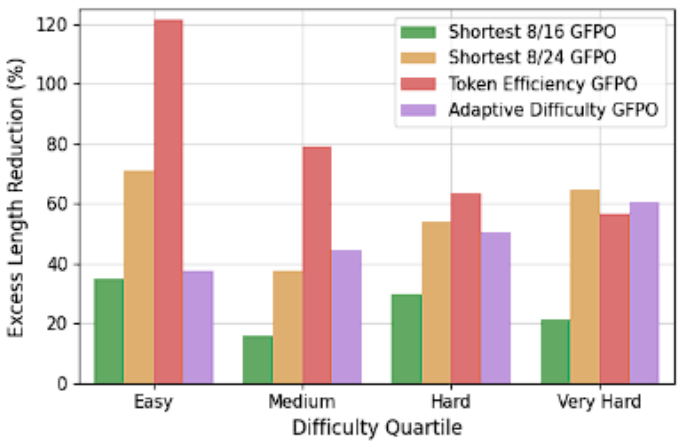
1. **仅依靠拒绝采样，不增大采样数量，能否减少思考**？观察G=8的实验中，长度压缩效果有限，而当G=16时，推理长度有了显著的下降。这验证了GFPO的结论，即必须在**训练时更多采样，才能实现推理时更少思考。**
2. **k 和 G对长度压缩的影响：**通过减少k或者增加G，都可以降低模型平均响应长度，保持 $\frac{k}{G}$ 在25%-33%的效果最好，进一步降低带来的增益会逐渐降低。



3. **Token Efficiency**：使用奖励除以长度作为过滤指标，优先保留短而正确的推理链，相比只保留最短k个响应的方法提供了更丰富的奖励，能更有效的抑制低价值的冗余token。缺点是训练过程中方差更高。
4. **自适应难度GFPO**：通过为不同难度的问题保留不同数量的响应，能更好的实现准确率和推理效率之间的平衡，在绝大多数baseline测试上都取得了最好的效果。
5. **分布外泛化能力**：在未经训练上，GRPO相比于SFT模型呈现出显出的长度膨胀，而准确率却没有提升；GFPO缓解了长度膨胀现象，准确率还略有提升。
6. **不同难度问题上的长度缩减**：随着问题难度上升，模型的响应长度有所增加，但GFPO实现了最好的长度压缩效果。Token Efficiency在简单问题上效果更好，自适应难度GFPO在最困难问题上效果更佳，有效缩减了长响应的“长尾”部分。

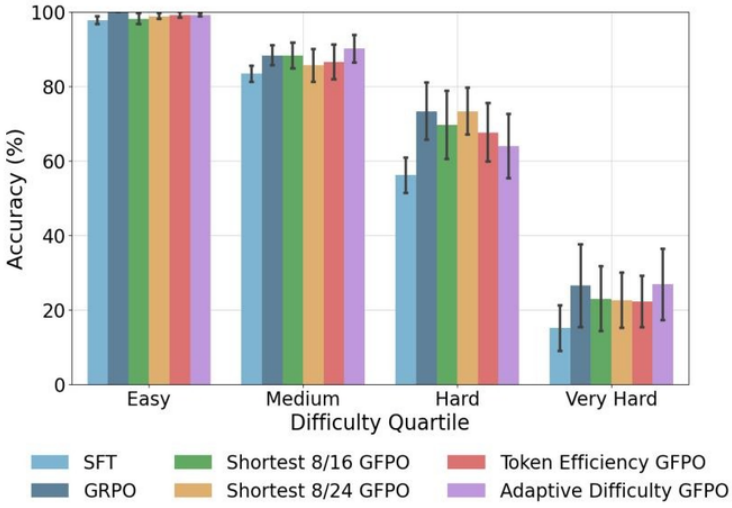


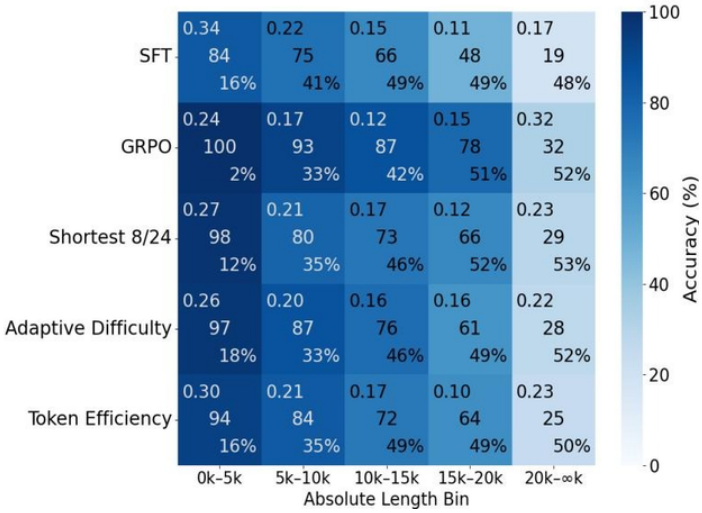
(a) Average Response Length Across Problem Difficulties.



(b) Excess Length Reduction Across Problem Difficulties.

7. **不同难度问题上的准确率**：随着问题难度的增加，RL后的模型显著优于SFT模型，自适应难度GFPO在简单、中等和非常困难的问题上都达到或超过了GRPO的准确率，同时大幅减少了推理长度，且组的规模越大，在难题上的准确性越高。





8. **准确率与推理长度的关系**：在难题和极难题上，准确率随着长度的增加而持续下降，推理的最优点大约为12-16k token，在这个区间内长度足够进行完整的推理，又不至于过度思考。而在最长的响应四分位数中，GFPO 的准确度优于 GRPO。
9. **GFPO裁剪了什么**？将推理链分解为问题、解题、验证、最终答案4部分，GRPO主要的长度再解题和验证两部分，GFPO则精准的对这两个阶段进行了压缩。