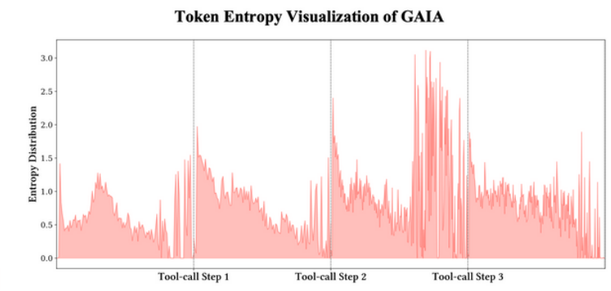
arxiv.org/abs/2507.19849

# AGENTIC REINFORCED POLICY OPTIMIZATION

## ARPO：面向TIR的基于entropy的自适应分支采样与策略优化方法

**Guanting Dong[1*], Hangyu Mao[2], Kai Ma[2], Licheng Bao[2*], Yifei Chen[1], Zhongyuan Wang[2*]**
**Zhongxia Chen[2], Jiazhen Du[2], Huiyang Wang[2*], Fuzheng Zhang[2], Guorui Zhou[2†]**
**Yutao Zhu[1], Ji-Rong Wen[1], Zhicheng Dou[1†]**
[1]Renmin University of China, [2]Kuaishou Technology
{dongguanting, dou}@ruc.edu.cn

GitHub: https://github.com/dongguanting/ARPO




## 简介

本文提出ARPO(Agentic Reinforced Policy Optimization)算法，用于在RLVR框架下训练具备TIR(tool-integrated reasoning)能力的lrm。简单来说，作者发现在TIR场景中每次拿到tool feedback之后，lrm token分布的entropy就很大，说明不太会思考了，于是针对这个问题，对GRPO进行了改进，ARPO引入基于token entropy的自适应rollout策略，如果lrm在调用工具后出现很不确定的思考那么就主动触发生成一条新的分支(branch)，目的是使用更多的算力来探索这个关键步骤，通过分支的方式来得到一组trajectory，此外，考虑到trajectory之间有共享token，又针对性的设计了advantage attribute estimation，显式或隐式的区分共享token与分支token，让优化更细粒度。

## 背景

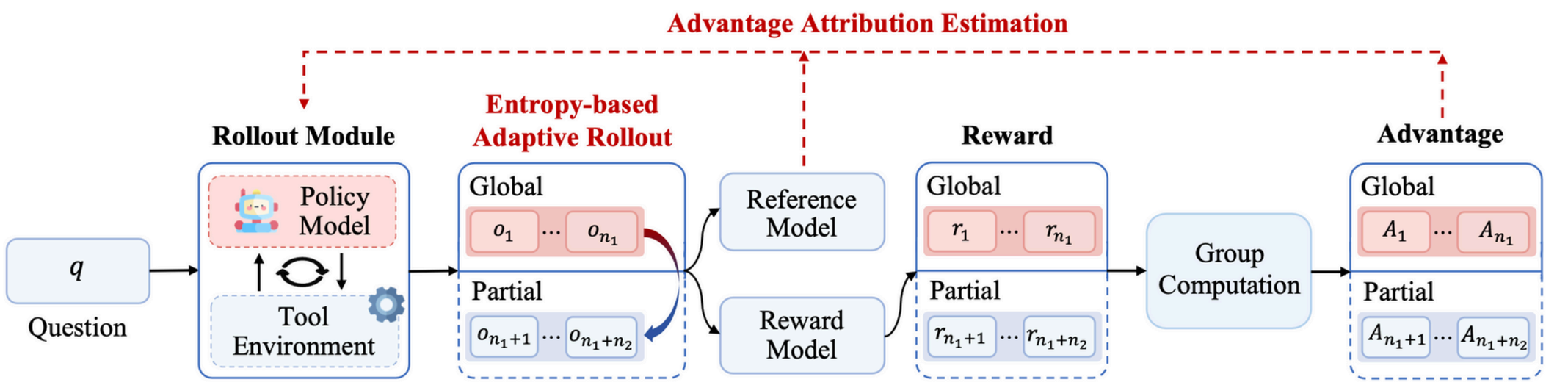
作者在使用GRPO算法遵循RLVR范式训练具备TIR(Tool-Integrated Reasoning)能力的lrm时，发现了一个问题：lrm在调用外部工具(如搜索引擎、代码解释器)之后，将工具的feedback拼接到reasoning trajectory继续decoding时，token分布会出现显著的entropy上升。说明lrm在拿到工具feedback之后由于数据差异，打破了原来的思考过程，对后续推理路径产生犹豫。为此，作者提出了ARPO(Agentic Reinforced Policy Optimization)算法，希望通过entropy驱动的自适应rollout机制来引导lrm更合理的使用工具。

## 实验设置

- 实验任务：数学推理、QA、Deep Search
- 三类tool：搜索引擎、web浏览器、代码解释器
- 对sft_llm做的实验

| Method | Mathematical Reasoning | | | | | Knowledge-Intensive Reasoning | | | | | Avg. |
|---|---|---|---|---|---|---|---|---|---|---|---|
| | AIME24 | AIME25 | MATH500 | GSM8K | MATH | WebWalker | HQA | 2Wiki. | MuSiQ. | Bamb. | |
| **Qwen2.5-3B-Instruct** | 10.0 | 6.7 | 63.0 | 75.0 | 71.6 | 0.5 | 9.7 | 9.4 | 3.6 | 11.7 | 26.1 |
| + TIR Prompting | 6.7 | 6.7 | 52.2 | 56.6 | 62.8 | 14.0 | 15.4 | 14.1 | 6.1 | 16.4 | 25.1 |
| + GRPO | 20.0 | 13.3 | **72.0** | **86.0** | 81.0 | 21.0 | 56.5 | 64.5 | 24.7 | 65.2 | 50.4 |
| + Reinforce ++ | 16.7 | 13.3 | 70.4 | 85.0 | 80.2 | 19.5 | 55.9 | 62.3 | 27.9 | 65.7 | 49.7 |
| + DAPO | 20.0 | 16.7 | 71.2 | 85.0 | 81.2 | 19.5 | 54.8 | 62.5 | **30.0** | 64.8 | 50.6 |
| + ARPO | **23.3** | **20.0** | 71.4 | 85.0 | **82.5** | **24.5** | **58.5** | **67.4** | 28.7 | **66.8** | **52.8** |
| **Llama3.1-8B-Instruct** | 3.3 | 0.0 | 43.3 | 81.4 | 60.6 | 3.0 | 24.3 | 24.6 | 10.4 | 40.0 | 28.8 |
| + TIR Prompting | 3.3 | 3.3 | 39.4 | 73.8 | 58.2 | 15.0 | 48.5 | 47.5 | 15.5 | 58.4 | 36.3 |
| + GRPO | 13.3 | 13.3 | 62.4 | 87.4 | 79.2 | 26.5 | 57.8 | 71.8 | 31.0 | 68.2 | 51.1 |
| + Reinforce ++ | 13.3 | **16.7** | 61.4 | 87.0 | 77.2 | 27.5 | 57.1 | 71.6 | 29.9 | 69.1 | 51.1 |
| + DAPO | 16.7 | 13.3 | 61.2 | 87.4 | 76.4 | 25.5 | 56.6 | 70.3 | 29.2 | 67.3 | 50.4 |
| + ARPO | **23.3** | **16.7** | **64.6** | **88.0** | **80.2** | **30.5** | **65.4** | **75.5** | **34.8** | **73.8** | **55.3** |
| **Qwen2.5-7B-Instruct** | 10.0 | 10.0 | 70.6 | 90.2 | 82.0 | 2.0 | 12.2 | 12.6 | 6.6 | 24.0 | 32.0 |
| + TIR Prompting | 6.7 | 10.0 | 68.2 | 64.6 | 78.2 | 15.5 | 14.8 | 18.3 | 9.5 | 23.6 | 31.0 |
| + GRPO | 23.3 | 26.7 | 78.0 | **92.8** | 87.8 | 22.0 | **59.0** | **76.1** | 30.6 | 68.4 | 56.5 |
| + Reinforce ++ | 26.7 | 23.3 | 78.0 | 92.2 | **88.8** | **26.0** | 55.1 | 68.9 | 25.2 | 64.9 | 54.9 |
| + DAPO | 20.0 | 23.3 | **80.4** | 91.0 | **88.8** | 24.0 | 57.7 | 68.4 | 28.6 | 65.5 | 54.8 |
| + ARPO | **30.0** | **30.0** | 78.8 | 92.2 | **88.8** | **26.0** | 58.8 | **76.1** | **31.1** | **71.5** | **58.3** |

## ARPO算法




ARPO基于GRPO同样需要一组trajectory，不同于GRPO独立的采样得到M条reasoning trajectory，ARPO将生成M条trajectory的过程融入到entropy-based adaptive rollout，通过在不确定的reasoning step处进行分支扩展新的trajectory，这样也能得到M条，同时还将更多的算力用在了不确定性较大的reasoning step。但是这样的问题是trajectory之间有共享token，为此又设计了advantage attribute estimation

## 思考

这篇论文的很大亮点在于作者对TIR训练过程中token entropy变化的细致观察，发现问题再去想办法解决问题。在TIR任务中，当lrm调用工具并收到反馈后，由于工具反馈数据和lrm思考过程数据的不一致，会短暂打破lrm的思考过程，这提醒我们，在融入工具反馈到reasoning trajectory的过程中，或许还有很多可以优化的地方，我们之前读过一篇对搜索引擎检索的文档做refinement后再append，也可以归属为这个方向。