ToolRL: Reward is All Tool Learning Needs

在RLVR背景下，如何设计PRM(PROCESS BASED REWARD)?

Cheng Qian, Emre Can Acikgoz, Qi He, Hongru Wang, Xiusi Chen, Dilek Hakkani-Tür, Gokhan Tur, Heng Ji
University of Illinois Urbana-Champaign
{chengq9, hengji}@illinois.edu

开源代码：HTTPS://GITHUB.COM/QIANCHENG0/TOOLRL

简介

本文提出了ToolRL，探索在非数学领域，延续RLVR做法，让llm学会使用tool(本文更像function calling)提升自己的能力，该如何设计reward function呢?

背景

目前LLM + Tool via RL的工作基本聚焦在数学领域，一方面是数学题很容易验证正确性，适合RLVR，另一方面相关的公开数据集非常丰富。本文则思考如何将LLM + Tool via RL扩展到通用领域，延续RLVR的做法，那么重点就是如何设计reward function?

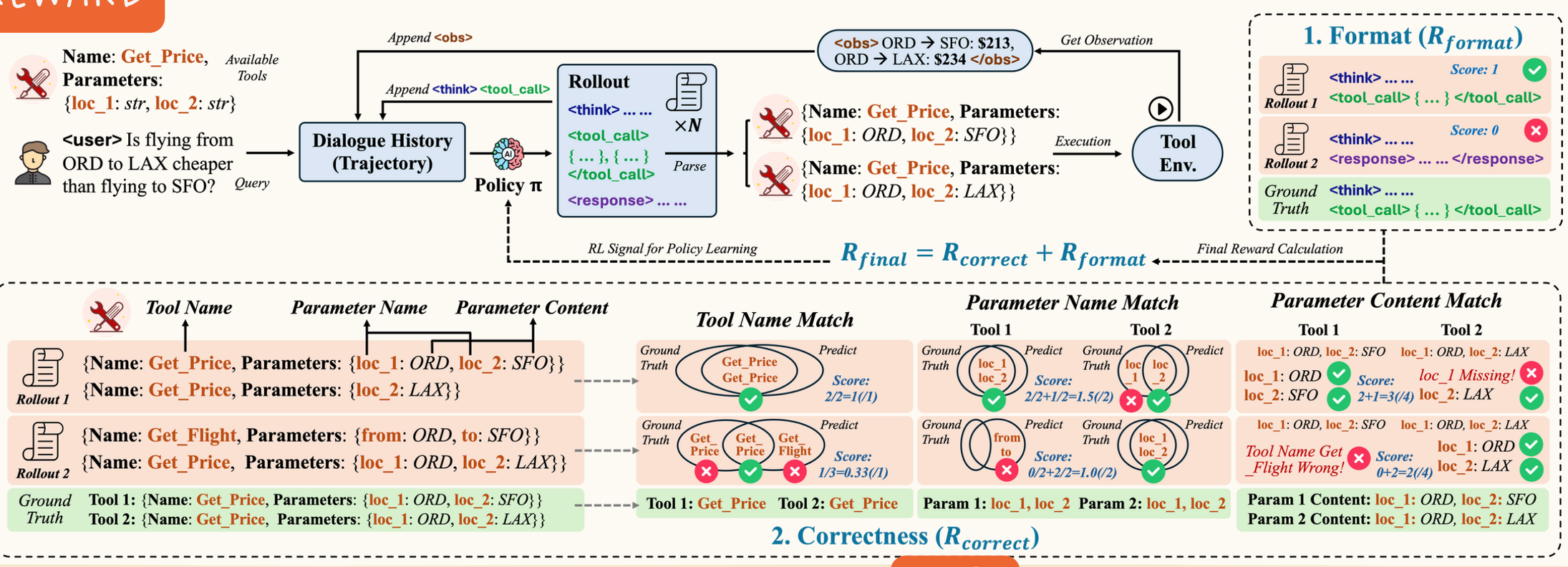
我个人认为，如果将数学领域的reward function类比为ORM(outcome based reward)吗，那么本文的reward function可以看作PRM。

实验设置

- 框架：verl
- 实验对象：Qwen-2.5-Instruct 和 Llama-3.2-Instruct
- 强化学习算法：GRPO
- 为了提升模型探索能力，去除了kl loss
- reward function：包含格式和function 调用正确性两部分

$$r_{\text{match}} = r_{\text{name}} + r_{\text{param}} + r_{\text{value}} \in [0, S_{\max}] \qquad \mathcal{R}_{\text{final}} = \mathcal{R}_{\text{format}} + \mathcal{R}_{\text{correct}} \in [-3, 4]$$

REWARD



部分实验结果

Model	Overall Acc	Non-Live AST Acc	Non-Live Exec Acc	Live Acc	Multi Turn Acc
Qwen2.5-1.5B-Instruct (**Raw**)	19.41%	16.00%	13.18%	35.58%	0.00%
Qwen2.5-1.5B-Instruct (**SFT400**)	40.21%	65.12%	61.11%	56.69%	1.00%
Qwen2.5-1.5B-Instruct (**SFT4k**)	40.67%	59.94%	59.84%	59.31%	1.00%
Qwen2.5-1.5B-Instruct (**SFT400+PPO**)	42.95%	77.65%	69.75%	55.73%	1.88%
Qwen2.5-1.5B-Instruct (**SFT400+GRPO**)	40.93%	70.54%	60.79%	56.33%	1.00%
Qwen2.5-1.5B-Instruct (**PPO Cold Start**)	38.32%	79.40%	70.11%	45.24%	0.87%
Qwen2.5-1.5B-Instruct (**Ours, GRPO Cold Start**)	**46.20%**	77.96%	76.98%	60.73%	2.25%
Qwen2.5-3B-Instruct (**Raw**)	33.04%	42.52%	40.80%	53.96%	1.00%
Qwen2.5-3B-Instruct (**SFT400**)	34.08%	69.29%	61.50%	41.40%	0.00%
Qwen2.5-3B-Instruct (**SFT4k**)	41.97%	62.85%	54.73%	59.17%	0.75%
Qwen2.5-3B-Instruct (**SFT400+PPO**)	45.80%	78.29%	71.09%	58.76%	5.12%
Qwen2.5-3B-Instruct (**SFT400+GRPO**)	46.42%	76.21%	68.93%	64.15%	1.75%
Qwen2.5-3B-Instruct (**PPO Cold Start**)	51.15%	82.42%	78.52%	67.78%	4.88%
Qwen2.5-3B-Instruct (**Ours, GRPO Cold Start**)	**52.98%**	81.58%	79.43%	73.78%	3.75%

思考

乍一看，本文和前面读过的TORL、ZTRL一个套路，但实际上差别还是很大的。数学领域的REWARD设计很简单，当然应用题和证明题除外，这也是为什么大家不用这两种题型训练模型。所以想要将RL + TOOL扩展到其他领域，难点是如何设计REWARD FUNCTION，当然前提是延续RLVR的做法，如果有REWARD MODEL那就是另一个话题了。因此关于数据集格式，会发现长这样：(PROMPT,GROUND_TRUTH_RESPONSE)，作者设计了细粒度的几个REWARD，去比较LLM生成的RESPONSE和GROUND_TRUTH_RESPONSE,我的第一感觉就是，这很像PRM啊。另外一点区别是，LLM直接生成完整的RESPONSE，包含了要调用的函数名称和参数值，而不是像TORL、ZTRL那样生成代码之后，还需要外部代码解释器执行，然后将执行结果返回给LLM，继续生成，因此我更偏向于用FUNCTION CALLING来表述，而非TOOL。此外可能是数据集自身的原因，REWARD 中竟然没有评估RESPONSE是否正确?