START: Self-taught Reasoner with Tools

如何用 HINT（提示）+ SFT 唤醒 QWQ 的 TOOL CALLING 能力?

Chengpeng Li[1,2]*, Mingfeng Xue[2]*, Zhenru Zhang[2], Jiaxi Yang[2]*, Beichen Zhang[2]*, Xiang Wang[1], Bowen Yu[2], Binyuan Hui[2], Junyang Lin[2], Dayiheng Liu[2]†

[1]University of Science and Technology of China

[2]Alibaba Group

{lichengpeng.lcp,liudayiheng.ldyh}@alibaba-inc.com

简介

本文提出了 START，一种让 QwQ“自学使用工具”的训练方法，核心流程分为两步：

Hint-infer：在原有的纯语言推理数据中插入自然语言提示，引导 QwQ 自动生成包含 Python 解释器调用的 TIR（Tool-Integrated Reasoning）数据。

Hint-RFT：对这些自生成数据进行打分筛选，再用 SFT 训练QwQ，最终得到内化了 Tool Calling 能力的强化版本 —— START 模型。

背景

随着LLM推理能力的增强，Long CoT逐渐成为主流：模型不仅给答案，还能一步步“思考”过程。但仅靠语言生成推理path往往不够，如果能再结合外部工具想必是极好的，这类方法被称为工具集成推理（Tool-Integrated Reasoning, TIR）。

目前主流做法是RL、SFT以及prompt，本文聚焦sft，一个自然的问题是如何得到TIR训练数据呢？START 提出了一种全新范式：不依赖人工标准数据，让模型“自学”如何使用工具，通过提示激发(Hint-Infer)、再微调强化(Hint-RFT)，最终获得稳定的 tool calling 能力。

实验

TOOL：PYTHON解释器

任务：数学推理和代码生成

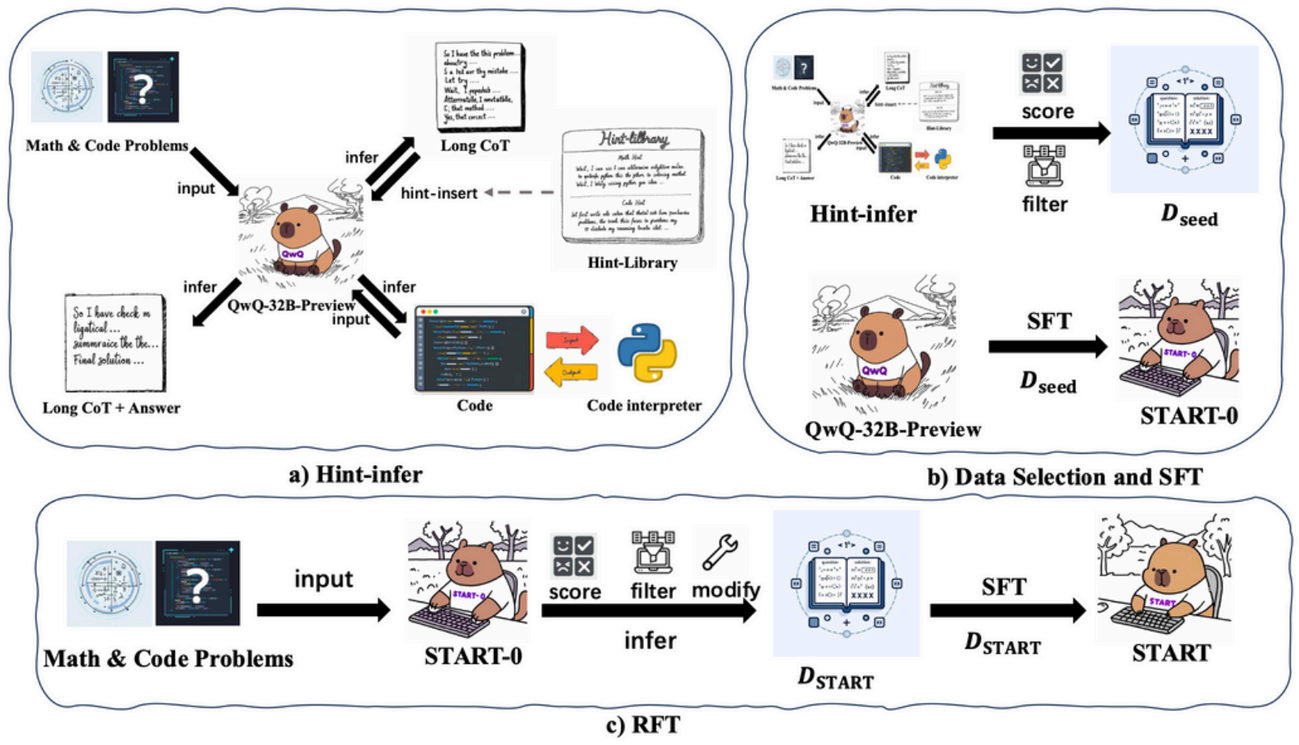
- 实验对象：QwQ-32B-preview
- 针对数学推理和代码生成，作者构造了很多hint(提示)，尽量覆盖方方面面，见下左图（Hint-Library）
- 如何使用hint?即hint如何插入到已有的纯语言推理path中：1）在高频连接词（比如“Alternatively”, “Wait”）之后插入 hint，因为这些词通常出现在模型“反思”、“切换思路”的时刻，插入hint让llm调用tool帮助自己；2）在generation即将结束时插入 hint，延长“思考时间”，鼓励补充 reasoning 和调用工具

HINT长什么样子?

Hint-Library

Code Hint	Math Hint
Debug hint Let me first write a code that includes all test cases from the problems to validate my reasoning locally. To ensure that my coderuns correctly, I need to embed all test case inputs directly into my code and print the corresponding output, following the sample structure below: ```python [A code template] ```output [...] Alright, with this structure, I can write and execute my code in a Python compiler using real example inputs. By comparing the actual outputs with the expected outputs, I can initially assess the correctness of my code. If the outputs do not match, I can debug accordingly. Recall the test cases in the problem statement.{testcase} Alright, now I can write a debug code with samples input. ```python	**Complex calculations hint** I can use Python to perform complex calculations for this problem.```python **Self-reflection hint** I can use Python to check if my approach is correct and refine it, if necessary.```python **Check logic hint** maybe Python can assist in ensuring our logical deductions are sound.```python **Alternative method hint** I can use Python to explore an alternative method for solving this problem.```python **General hint** maybe using python here is a good idea.```python **Deeper think hint** I can think more deeply about this problem through python tools.```python
(A hint for code question without starter code)	(Different Functional hints for math question)

训练流程



部分实验结果

Method	GPQA	MATH500	AMC23	AIME24	AIME25	LiveCodeBench
			General LLMs			
Qwen2.5-32B	46.4	75.8	57.5	23.3	-	22.3
Qwen2.5-Coder-32B	33.8	71.2	67.5	20.0	-	25.0
Llama3.3-70B	43.4	70.8	47.5	36.7	-	34.8
DeepSeek-V3-671B	59.1	90.2	-	39.2	-	40.5
GPT-4o†	50.6	60.3	-	9.3	-	33.4
			Reasoning LLMs			
API Only						
o1-preview†	73.3	85.5	81.8	44.6	37.5	53.6
o1-mini†	-	90.0	-	63.6	50.8	-
o1†	77.3	94.8	-	74.4	-	63.4
o3-mini(low)†	70.6	95.8	-	60.0	44.2	75.6
Open weights						
R1-Distill-Qwen-32B†	62.1	94.3	93.8	**72.6**	46.7	**57.2**
s1-32B†	59.6	93.0	-	50.0	33.3	-
Search-o1-32B†	**63.6**	86.4	85.0	56.7	-	33.0
QwQ-32B-Preview	58.1	90.6	80.0	50.0	40.0	41.4
START	**63.6**(+5.5)	**94.4**(+3.8)	**95.0**(+15.0)	66.7(+16.7)	**47.1**(+7.1)	47.3(+5.9)

思考

让QWQ自己生成数据SFT自己，然后还能提升TIR能力，感觉好矛盾啊，如何解释呢？作者的意思是QWQ实际上已经具备调用工具的能力，只是这种能力在原始模型中处于潜在（LATENT）状态，通过 HINT-INFER 的方式可以被激活，而 HINT-RFT 则进一步将这种能力内化。 从实验结果看，用HINT-INFER可以提升QWQ的TIR能力，再经过SFT这种能力更强了，说明“提示能唤醒，训练能稳固”是有效的策略。

这也让我联想到近期大量使用 RLVR 声称“让 QWEN2.5 学会了某种能力”的工作。在采信这些结论之前，我们或许应该追问一下：这些被“学会”的能力，是模型原本就具备的，只是被 RL 引导激活出来？还是通过 RL 真正学习到了全新知识？