

# 基于verl-retool的昇腾Agentic RL训练实践

作者：周尖

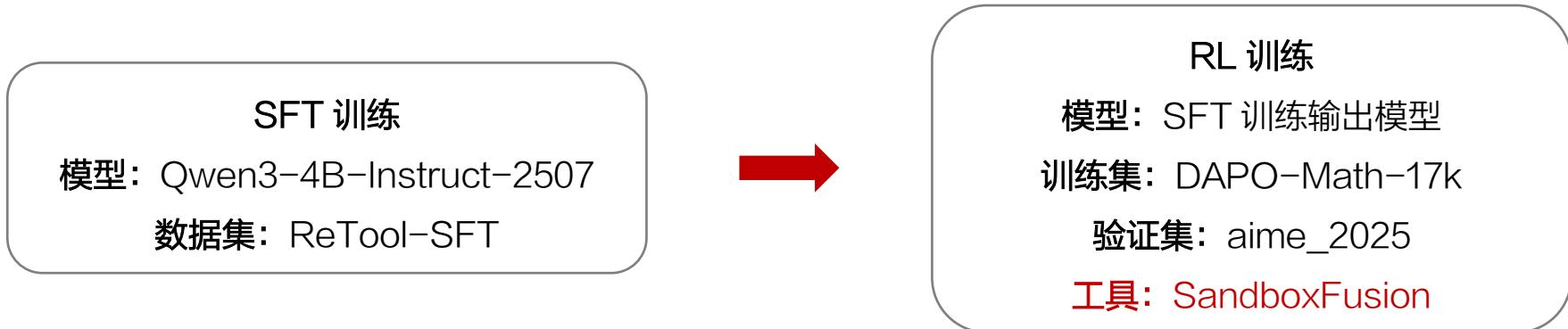
时间：2025.12.18

<https://gitcode.com/cann>

**CANN**

# 样例介绍

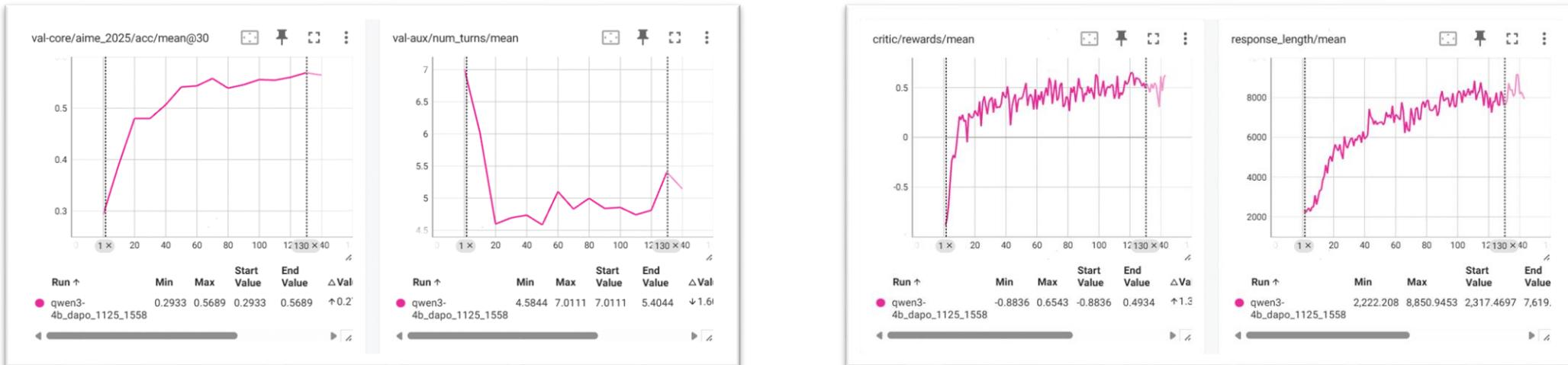
- ✓ 本样例参考verl/recipe中的retool项目，在NPU上进行agent RL训练。训练过程分为如下两步：
  - SFT训练：使用ReTool-SFT数据集对Qwen3-4B-Instruct-2507模型进行SFT训练，帮助模型学会调用Sandbox工具进行python编程；
  - RL训练：使用DAPO-Math-17k数据集对前一步SFT过的模型进行RL训练，帮助模型更好地使用Sandbox工具，辅助自身推理思考。



# 样例结果

✓ 训练效果：通过RL训练，模型在测试集上的准确率从0.29提高至0.57。

model	step	val_dataset	accuracy	num_turns
Qwen3-4B-Instruct-2507-sft	0	aime_2025	0.29	7.0
Qwen3-4B-Instruct-2507-rl	130	aime_2025	0.57	5.4



➤ Tensorboard可视化训练中关键指标的变化

# response样例

## ✓ 输出样例说明：

输出样例分为四个关键部分：

- prompt ( 模型输入 )
- response ( 模型输出-文本输出 )
- tool call ( 模型输出-工具调用 )
- tool response ( 工具响应 )

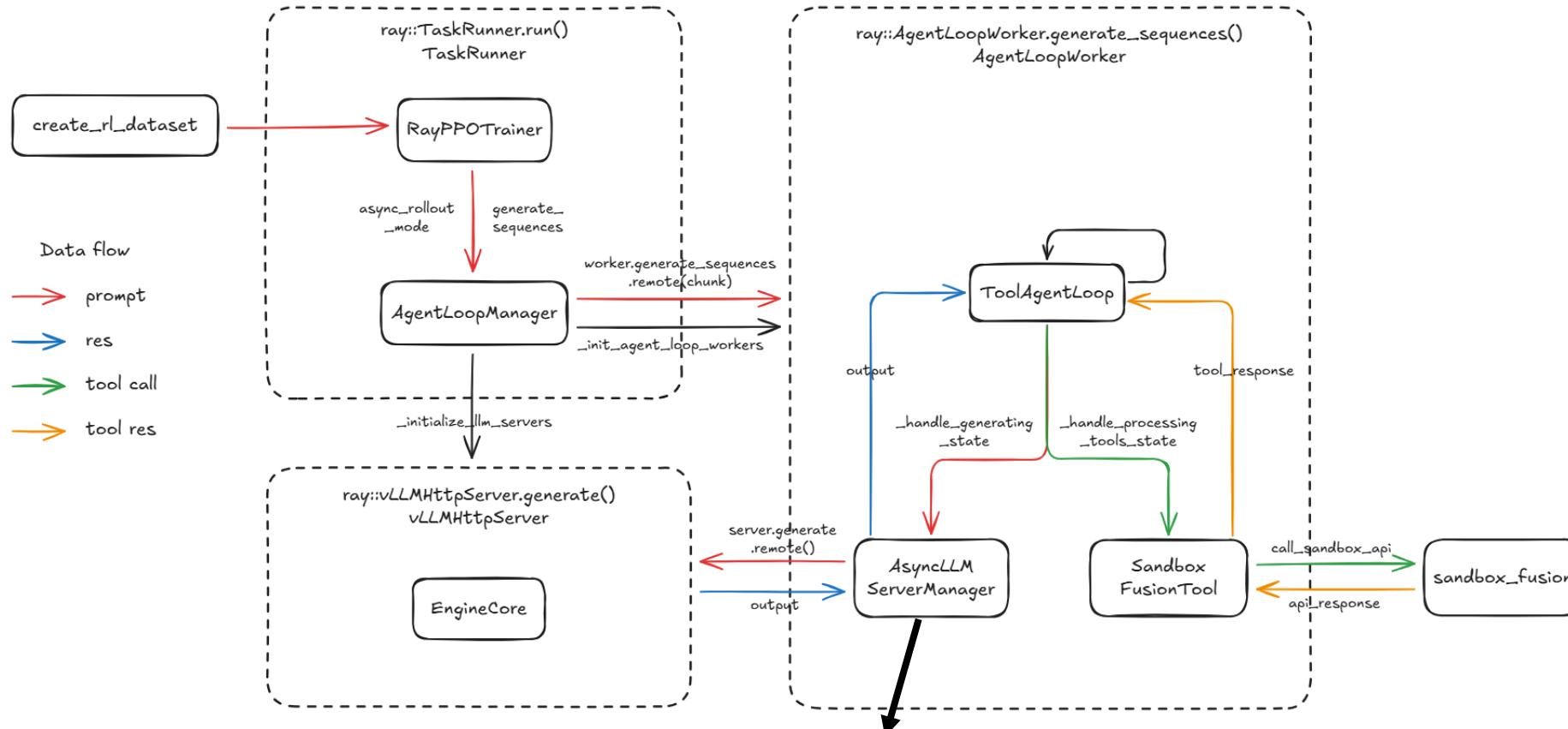
问题输入→模型思考→调用代码工具  
→返回结果作为新一轮的输入→模型

思考→生成答案

<b>输出样例</b>
<b>prompt</b>
<b>user</b>
Sixteen chairs are arranged in a row. Eight people each select a chair in which to sit so that no person sits next to two other people. Let \$N\$ be the number of subsets of the 16 chairs that could be selected. Find the remainder when \$N\$ is divided by \$1000\$.
The answer format must be: \boxed{The final answer goes here.}
<b>response</b>
<b>assistant</b>
Okay, so there's this problem where sixteen chairs are arranged in a row, and eight people each select a chair. The condition is that no person sits next to two other people. ... Therefore, I need to compute the number of such binary strings. How do we compute that? ... But this might be a bit tedious, but since we need to compute up to n=16 and k=8, perhaps we can use code to compute this.
<b>tool call</b>
<tool_call>
{ "name": "code_interpreter", "arguments": { "code": "def compute_dp():\n    # Initialize a 3D DP array: dp[n][k][c], where n from 0 to\n    16, k from 0 to 8, c from 0 to 2\n    dp = [[[0 for _ in range(3)] for _ in range(9)] for _ in range(17)]\n    \n    # Base case: n=0,\n    k=0\n    dp[0][0][0] = 1\n    \n    for n in range(1, 17):\n        for k in range(0, 9):\n            # Transition for c=0: adding a 0 to any state with same k\n            dp[n][k][0] = dp[n-1][k][0] + dp[n-1][k][1] + dp[n-1][k][2]\n            \n            # Transition for c=1: adding a 1 when k>=1\n            if k >= 1:\n                dp[n][k][1] = dp[n-1][k-1][0]\n            else:\n                dp[n][k][1] = 0\n                \n                # Transition for c=2: adding a 1 when k>=1 and previous was c=1\n                if k >= 1:\n                    dp[n][k][2] = dp[n-1][k-1][1]\n                else:\n                    dp[n][k][2] = 0\n            \n            return dp\n        \n        dp = compute_dp()\n        result = dp[16][8][0] + dp[16][8][1] + dp[16][8][2]\n        print(result % 1000)\n    }\n}\n</tool_call>
<b>tool response</b>
<b>user</b>
<tool_response>
907
</tool_response>
<b>response</b>
<b>assistant</b>
Therefore, the number of subsets N is 907 mod 1000. Wait, but let me verify this code. The code initializes a 3D array where dp[n][k][c] represents the number of binary strings of length n with k 1s, ending with c consecutive 1s. The transitions are set according to the recurrence relations. The final result is the sum over c=0,1,2 for k=8 and n=16. The code returns 907 modulo 1000. This matches the previous manual calculation. Therefore, the answer is 907. ... Therefore, the remainder when N is divided by 1000 is \boxed{907}.
\boxed{907}

# ToolAgent代码结构

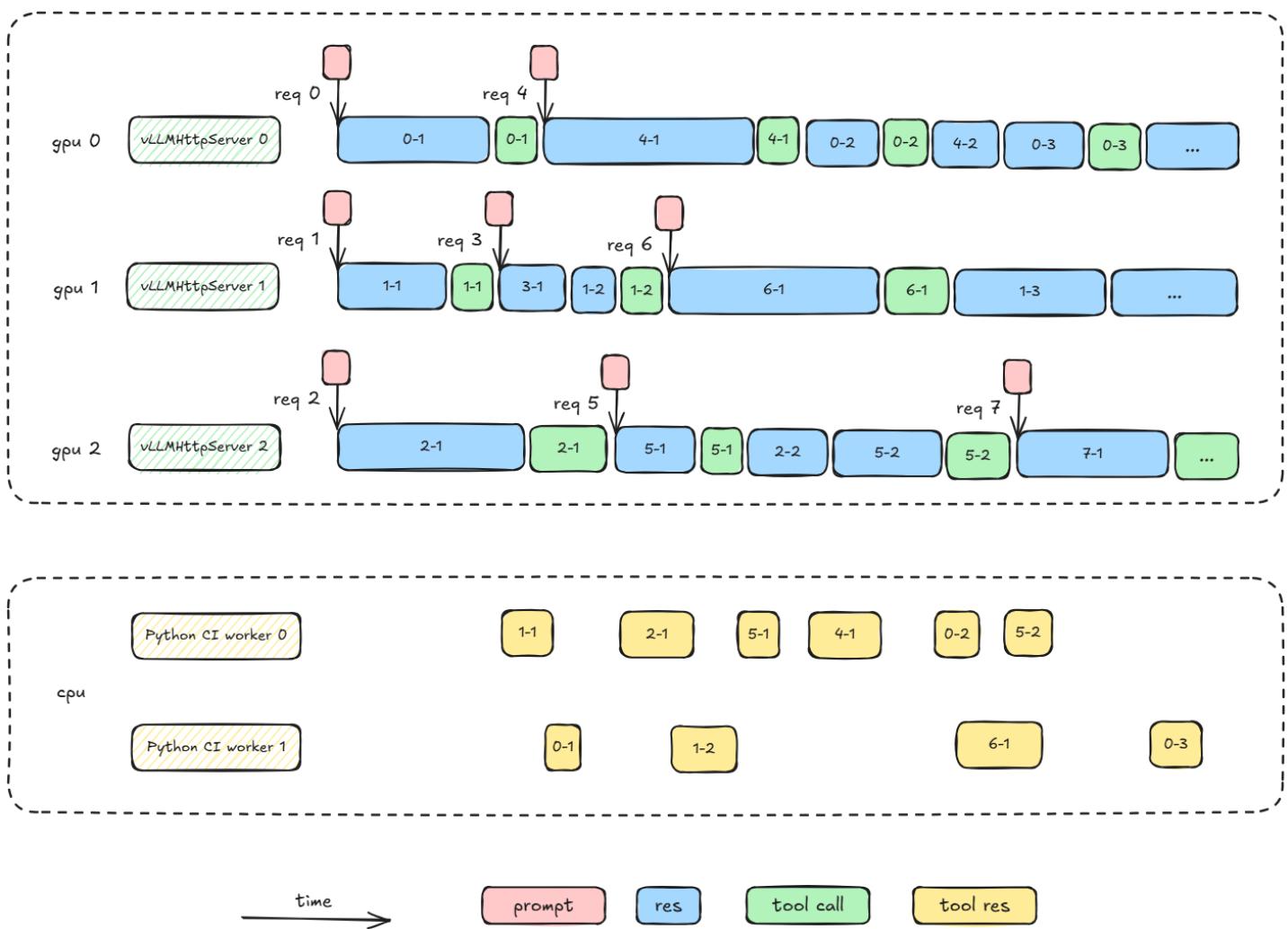
- ✓ ToolAgentLoop是agent RL训练中的核心类，是实现multi-turn交互中Agent reasoning→tool call循环的关键。



- 负载均衡：首轮请求选择请求最少的Server
- 粘性会话：多轮会话共用一个Server

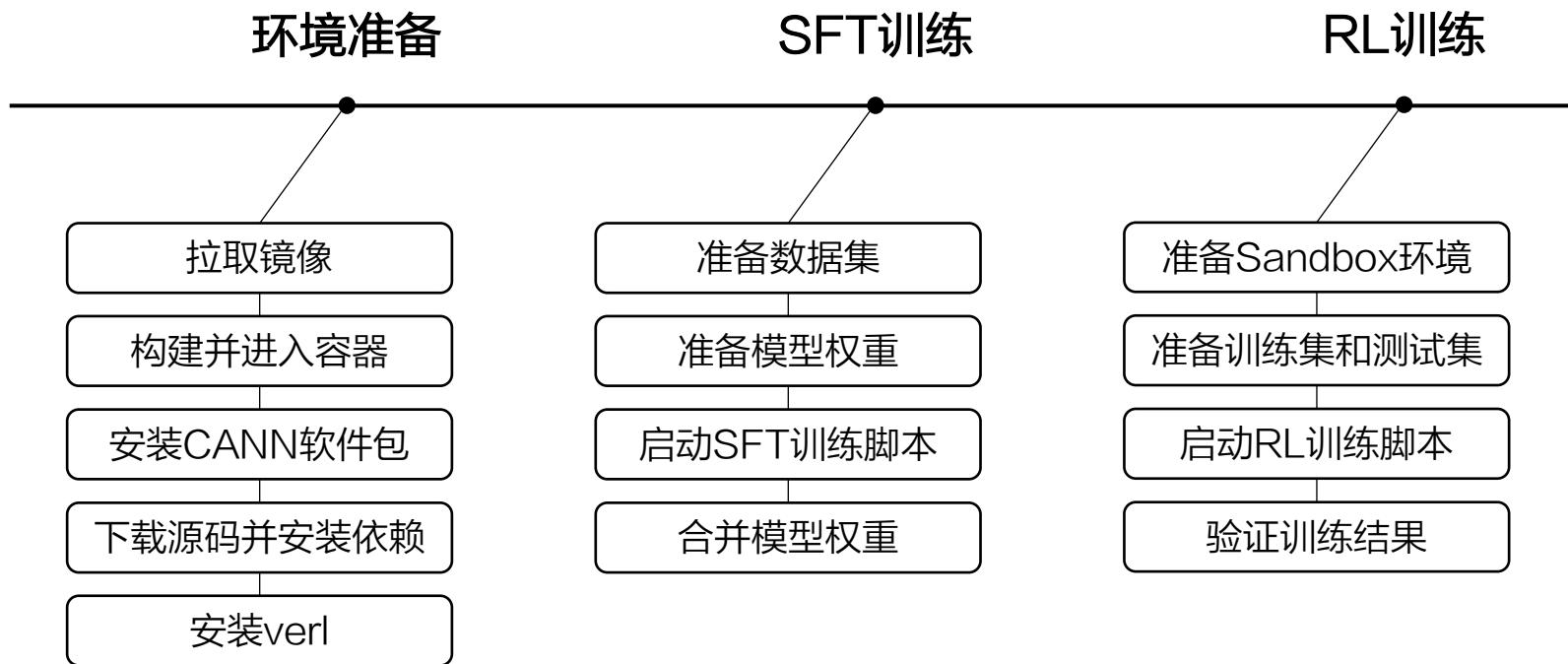
# ToolAgent异步流水

- ✓ 在ToolAgent异步流水中，每一个prompt的request都会遵循AsyncLLMServerManager的规则，在此基础上实现vLLM推理和Python CI工具调用的异步并行。



# 样例复现

✓ 跟随Readme，丝滑复现agent RL训练：



## 目录

- 基于verl-retool的agent样例
  - 概述
  - 支持的产品型号
  - 文件说明
  - 环境准备
  - SFT训练
  - RL训练
  - 附录
    - ToolAgent代码结构
    - ToolAgent异步流水



欢迎到仓库提交issue/PR



欢迎通过SIG联系我们

<https://gitcode.com/cann>

**CANN**

# Thank you.

社区愿景：打造开放易用、技术领先的AI算力新生态

社区使命：使能开发者基于CANN社区自主研究创新，构筑根深叶茂、跨产业协同共享共赢的CANN生态

Vision: Building an Open, Easy-to-Use, and Technology-leading AI Computing Ecosystem

Mission: Enable developers to independently research and innovate based on the CANN community and build a win-win CANN ecosystem with deep roots and cross-industry collaboration and sharing.



上CANN社区获取干货



关注CANN公众号获取资讯