

PR #2368 完整报告

radixark/miles

fix(rollout): group session v2 leaf samples

合并时间: 2026-08-12 11:39

原文链接: <http://prhub.com.cn/radixark/miles/pull/2368>

执行摘要

- 一句话: session v2 多叶子样本共享 rollout_id 并补失败保护
- 推荐动作: 值得精读, 改动虽小但触及 session v2 数据契约的关键环节。建议重点关注两处设计: 一是 rollout_id 的优先 / 回退策略与 FIXME 背后的 index 语义问题; 二是用 assert 做前置校验在优化模式下的失效风险, 后续可替换为显式异常。测试中用真实下游校验函数 validate_compact_rollout_ids 验证产物是好实践, 值得在其他生成路径推广。

功能与动机

PR body 明确说明: session server v2 可能为一次环境 rollout 返回多个可训练叶子, 包括 compaction 或子代理创建的叶子; 这些叶子需要共享 rollout 身份, Miles 才能校验嵌套输出、应用 rollout 级掩码, 并在下游奖励处理中把彼此视为兄弟样本。原实现未对 v2 叶子做身份统一, 可能发出无法安全分组的样本, 因此需要尽早失败并回填身份。

实现拆解

1. 在 miles/rollout/generate_hub/agent_tool_call.py 的 generate() 中, 于 use_v2 分支拿到 result.samples 之后新增身份推导逻辑: rollout_id = input.sample.rollout_id if input.sample.rollout_id is not None else input.sample.index, 优先保留显式传入的 rollout_id, 缺失时回退到全局分配的 Sample.index。
2. 增加早期失败保护: 若两者均为 None, 立即 assert rollout_id is not None 抛出 AssertionError, 确保无身份样本不会继续流向下流; 随后将推导出的 rollout_id 统一赋给所有 v2 叶子样本, 使 compaction 或子代理产生的兄弟叶子共享同一身份。
3. 在 tests/fast/rollout/generate_hub/test_agent_v2.py 中新增两个测试: test_success_assigns_shared_rollout_id_to_v2_leaves 用参数化覆盖显式 id (11 原样保留) 与 fallback (index=7) 两种路径, 并调用下游真实校验函数 validate_compact_rollout_ids 验证产物可直接进入 compact rollout 校验流程; test_v2_requires_input_rollout_identity 验证 index 与 rollout_id 同时为 None 时抛出带指定消息的 AssertionError。
4. 配套说明: PR body 明确本次不新增用户侧选项、不更新 README, 仅补齐既有 Sample.rollout_id 契约; 第二个提交由合并者 guapisolo 直接更新源码文件, 可能用于微调断言细节。

关键文件:

- miles/rollout/generate_hub/agent_tool_call.py (模块 生成中心; 类别 source; 类型 core-logic; 符号 generate) : 核心修复所在: 在 generate() 的 v2 分支统一为所有返回叶子回填 rollout_id, 并增加无身份时的早期失败保护, 直接影响下游样本分组与校验。
- tests/fast/rollout/generate_hub/test_agent_v2.py (模块 会话测试; 类别 test; 类型 test-coverage; 符号 test_success_assigns_shared_rollout_id_to_v2_leaves, test_v2_requires_input_rollout_identity) : 新增两个测试固化 v2 叶子身份行为: 显式 id 保留、index 回退、无身份时立即失败, 并用下游真实校验函数验证产物可用性。

关键符号: generate, test_success_assigns_shared_rollout_id_to_v2_leaves, test_v2_requires_input_rollout_identity

关键源码片段

miles/rollout/generate_hub/agent_tool_call.py

核心修复所在: 在 generate() 的 v2 分支统一为所有返回叶子回填 rollout_id, 并增加无身份时的早期失败保护, 直接影响下游样本分组与校验。

```
# 收集阶段: v2 走 `collect_samples`, agent_metadata 由服务端合并回样本。
if collect_timed_out:
    sample = deepcopy(input.sample)
    sample.status = Sample.Status.ABORTED
    return GenerateFnOutput(samples=[sample] if use_v2 else sample)

if not result.samples:
    # 空回复或全被截断时, 产出 ABORTED 样本, 保证调用方拿到确定性的列表形态。
    sample = deepcopy(input.sample)
    sample.status = Sample.Status.ABORTED
    return GenerateFnOutput(samples=[sample] if use_v2 else sample)

samples = result.samples
if use_v2:
    # FIXME: handle sample index issues. —— 输入 Sample.index 的全局分配语义尚未完全保证。
    # 优先使用显式传入的 rollout_id; 若缺省则回退到全局分配的 Sample.index。
    # 这样 compaction
    # 或子代理派生的多个叶子样本才能共享同一身份, 供下游校验、掩码与奖励处理使用。
    rollout_id = input.sample.rollout_id if input.sample.rollout_id is not None else input.sample.index
    assert rollout_id is not None, "v2 agentic samples require input Sample.rollout_id or Sample.index"
    for sample in samples:
        sample.rollout_id = rollout_id
if not use_v2:
    # v1: agent 元数据在 driver 侧覆盖; v2 已由服务端合并, 无需再次覆盖。
    for s in samples:
        s.metadata.update(agent_metadata or {})

# 把 agent 在环境 / 工具步骤中消耗的墙钟时间反映到 Sample.non_generation_time, 供吞吐统计扣除。
```

```

ngt = ((agent_metadata or {}).get("agent_metrics") or {}).get("total_tool_time")
if ngt is not None:
    for s in samples:
        s.non_generation_time = ngt

if use_v2:
    return GenerateFnOutput(samples=samples)

(sample,) = samples
sample.metadata.update(result.session_metadata)
return GenerateFnOutput(samples=sample)

```

tests/fast/rollout/generate_hub/test_agentic_v2.py

新增两个测试固化 v2 叶子身份行为：显式 id 保留、index 回退、无身份时立即失败，并用下游真实校验函数验证产物可用性。

```

@pytest.mark.asyncio
@pytest.mark.parametrize(("input_rollout_id", "expected_rollout_id"), [(None, 7), (11, 11)])
async def test_success_assigns_shared_rollout_id_to_v2_leaves(monkeypatch, input_rollout_id,
expected_rollout_id):
    # 模拟 v2 返回两个叶子样本（例如 compaction 或子代理产生），
    # 二者必须被分配同一个 rollout_id，才能作为同一个 rollout 的兄弟样本参与下游处理。
    leaves = [
        Sample(status=Sample.Status.COMPLETED, response="one", response_length=1, tokens=
[1]),
        Sample(status=Sample.Status.COMPLETED, response="two", response_length=1, tokens=
[2]),
    ]
    tracer = _Tracer(SamplesReply(samples=leaves, session_metadata={}, empty_reason=None))
    _patch_agent(monkeypatch, tracer)
    generate_input = _generate_input()
    generate_input.sample.rollout_id = input_rollout_id

    output = await agentic_tool_call.generate(generate_input)

    # 显式 rollout_id 原样保留；缺省时回退到 Sample.index（测试里设为 7）。
    assert [sample.rollout_id for sample in output.samples] == [expected_rollout_id] * 2
    # 用下游真实校验函数确认输出可直接进入 compact rollout 校验流程。
    validate_compact_rollout_ids([[output.samples]])

```

```

@pytest.mark.asyncio
async def test_v2_requires_input_rollout_identity(monkeypatch):
    # 既没有 rollout_id 也没有 index 时，应当在生成阶段立刻失败，
    # 而不是把无法安全分组的样本继续往下游发送。
    leaf = Sample(status=Sample.Status.COMPLETED, response="done", response_length=1,
tokens=[1])
    tracer = _Tracer(SamplesReply(samples=[leaf], session_metadata={}, empty_reason=None))
    _patch_agent(monkeypatch, tracer)

```

```
generate_input = _generate_input()
generate_input.sample.index = None
generate_input.sample.rollout_id = None
```

```
with pytest.raises(AssertionError, match="require input Sample.rollout_id or Sample.index"):
    await agentic_tool_call.generate(generate_input)
```

评论区精华

本 PR 没有独立的 review 评论或讨论线程。唯一的审核意见来自合并者 guapisolo 的 APPROVE，评论为 "good fix."，未引发任何设计争议。两个提交中，第二个提交由 guapisolo 直接修改 `agentic_tool_call.py`，推测是对首个提交中身份推导或断言细节的补充微调。

- 暂无高价值评论线程

风险与影响

- 风险：

1. `assert` 在 Python -O 优化模式下会被剥离，若生产环境启用优化，无身份样本可能绕过失败保护继续流出；建议后续改为显式抛出 `ValueError` 或类似异常。
2. 源码中留有 `# FIXME: handle sample index issues` 注释，说明 `Sample.index` 的全局分配语义尚未完全保证；若 `compaction` 路径复用或清空 `index`，可能导致不同 `rollout` 被误分组，当前断言只能兜底 `None` 场景。
3. 对 v2 叶子统一覆盖 `rollout_id` 会覆盖服务端已返回的 `id`（若有）；当前需求恰好需要统一，但若未来服务端分配更有意义的 `id`，该行为需要重新评估并补充文档。
4. v1 路径与空回复、超时分支完全不受影响，因为赋值逻辑只在 `result.samples` 非空且 `use_v2` 为真时执行。
 - 影响：对用户与开发者无新增配置项，行为透明；对系统而言，`session v2 agentic rollout` 产生多个叶子时，嵌套输出校验、`rollout` 级掩码和下游奖励处理现在可以按组正确聚合，并与 `reward-normalization` 配套 PR #2369 协同工作。对团队而言，该变更确立了 v2 叶子的身份契约（显式 `rollout_id` 优先、`index` 兜底），并用两个测试固化，降低了后续数据管道回归风险。影响面集中在 `agentic v2` 生成路径，代码量小，风险可控。
 - 风险标记：核心数据契约路径变更，`assert` 校验可能被优化移除，`rollout_id` 覆盖服务端已有值，`index` 回退语义待澄清（`FIXME`）

关联脉络

- PR #2369 `reward-normalization companion PR`: PR body 明确说明本变更与 `reward-normalization` 配套 PR #2369 相互独立、同期合入主分支，二者共同完善 `session v2` 下游样本处理（奖励归一化与身份分组）。