PR #6251 完整报告

verl-project/verl

[trainer] fix: write request_id to reward_extra_infos_to_dump instead of reward_extra_infos_dict

合并时间：2026-05-07 11:50

原文链接：http://prhub.com.cn/verl-project/verl/pull/6251

执行摘要

- 一句话：修复 request_id 写入错误字典的问题
- 推荐动作：值得合入。该 PR 修复了一个明确的 bug，修复方式直接且正确。虽然改动小，但对依赖 request_id 进行下游分析的用户很重要。建议合入后关闭关联 Issue #6250。

功能与动机

Issue #6250 报告了一个 bug：在 _log_rollout_data 中，request_id 被写入 reward_extra_infos_dict 而不是 reward_extra_infos_to_dump。由于后者在插入 request_id 之前已从前一个字典创建并传递给 _dump_generations，导致转储的 rollout 数据中始终缺少 request_id，影响多轮 /agent-loop 场景中的轨迹追踪。

实现拆解

1. 定位问题：在 verl/trainer/ppo/ray_trainer.py 的 _log_rollout_data 方法中，第 452 行将 request_id 通过 setdefault 写入 reward_extra_infos_dict。
2. 分析根因：reward_extra_infos_to_dump 在第 448-450 行从 reward_extra_infos_dict 创建（此时还不包含 request_id），随后在第 462 行被传递给 _dump_generations。因此，无论 request_id 是否被写入源字典，转储数据中都不会包含它。
3. 应用修复：将第 452 行的字典引用从 reward_extra_infos_dict 改为 reward_extra_infos_to_dump，这样 request_id 会直接被添加到即将转储的字典中。
4. 验证：通过代码审查确认修复正确，无需额外测试。

关键文件：

- verl/trainer/ppo/ray_trainer.py（模块 训练器；类别 source；类型 core-logic；符号 _log_rollout_data）：该文件是 PPO ray trainer 的入口，其中 _log_rollout_data 方法负责将 rollout 数据（包括 request_id）转储到磁盘。本 PR 修复了该方法的变量引用错误。

关键符号：_log_rollout_data

关键源码片段

verl/trainer/ppo/ray_trainer.py

该文件是 PPO ray trainer 的入口，其中 _log_rollout_data 方法负责将 rollout 数据（包括 request_id）转储到磁盘。本 PR 修复了该方法的变量引用错误。

```
# verl/trainer/ppo/ray_trainer.py ( 修复后 )

def _log_rollout_data(
    self, batch: DataProto, reward_extra_infos_dict: dict, timing_raw: dict, rollout_data_dir: str
):
    """Log rollout data to disk."""
    with marked_timer("dump_rollout_generations", timing_raw, color="green"):
        inputs = self.tokenizer.batch_decode(batch.batch["prompts"], skip_special_tokens=True)
        outputs = self.tokenizer.batch_decode(batch.batch["responses"], skip_special_tokens=True)
        scores = batch.batch["token_level_scores"].sum(-1).cpu().tolist()
        sample_gts = [item.non_tensor_batch.get("reward_model", {}).get("ground_truth", None)
        for item in batch]

        # 先基于 reward_extra_infos_dict 创建转储字典副本
        reward_extra_infos_to_dump = {
            k: (v.tolist() if isinstance(v, np.ndarray) else v) for k, v in reward_extra_infos_dict.
            items()
        }
        if "request_id" in batch.non_tensor_batch:
            # 修复：将 request_id 写入转储字典，而非源字典。
            # 之前写入 reward_extra_infos_dict 导致 request_id 永不被包含在转储输出中。
            reward_extra_infos_to_dump.setdefault(
                "request_id",
                batch.non_tensor_batch["request_id"].tolist(),
            )

        self._dump_generations(
            inputs=inputs,
            outputs=outputs,
            gts=sample_gts,
            scores=scores,
            reward_extra_infos_dict=reward_extra_infos_to_dump, # 传递给转储函数
            dump_path=rollout_data_dir,
        )
```

评论区精华

该 PR 无实质性 review 讨论。仅有的自动化检查无反馈，维护者 wuxibin89 直接审批通过。关联 Issue #6250 中额外提到一个潜在问题：如果 `reward_extra_infos_dict` 已包含 `request_id` 键（例如来自奖励计算），`setdefault` 将不会覆盖，`batch.non_tensor_batch["request_id"]` 的值会被静默丢弃。这可能值得单独审查。

- 暂无高价值评论线程

风险与影响

- 风险：低风险。修复仅涉及一行变量名更改，将写入的目标字典从源字典改为转储字典。逻辑上正确且无副作用。唯一潜在的边缘情况是如果 reward_extra_infos_dict 本身已包含 request_id 键（来自奖励计算），则转储数据中的 request_id 将来自源字典而非

batch.non_tensor_batch，但这与修复前行为一致（修复前该字段缺失），因此不增加新风险。

- 影响：影响范围：仅影响使用 rollout_data_dir 进行生成数据转储的用户。修复后，转储的 JSON 文件中将包含 request_id 字段，使得在多轮 /agent-loop 训练场景中能够正确追踪轨迹。对于未依赖 request_id 的场景无影响。修复行仅 1 行，不会引入回归。
- 风险标记：缺少测试覆盖

关联脉络

- 暂无明显关联 PR