PR #2498 完整报告

radixark/miles

fix: require shared rewards within rollouts

合并时间：2026-08-13 11:44

原文链接：http://prhub.com.cn/radixark/miles/pull/2498

执行摘要

- 一句话：修复 rollout 内 sibling reward 不一致被静默降级的问题
- 推荐动作：值得精读，尤其是“用硬校验替代启发式”的 reward 数据契约设计。对维护者，建议确认所有 rollout 数据源（session v2、legacy fixed-fanout）都满足 sibling 奖励一致契约；对使用者，若训练突然报 must share one reward，应检查 reward 后处理是否按 rollout 输出同一值，而不是依赖 mask 或长度差异。

功能与动机

PR body 指出：Siblings with different rewards were silently reduced to the reward of the leaf with the most trainable tokens, so masking or leaf length could change the rollout advantage。根因是 _normalize_rewards_by_rollout 分组时未要求共享 selected reward，_trainable_token_count 用最终训练 mask 和 response length 排名，mainstream_segments 静默采用排名 leaf 作为归一化输入。作者认为 rollout sibling rewards 是同一轨迹级信号，不应因 mask 或 leaf 长度影响 advantage，因此用显式校验替代静默启发式。

实现拆解

1. 删除启发式排序函数：在 miles/ray/rollout/train_data_conversion.py 中删除 _trainable_token_count，它按 remove_sample 与 effective_response_length 给 sibling 排名，是 mainstream 启发式的依据。
2. 重写 _normalize_rewards_by_rollout 的组内逻辑：rollout_segment_groups 由 values() 改为 items() 保留 rollout 键；每个 rollout 提取其所有 segment 的 raw_rewards，若存在与第一个奖励不同的值，抛出包含 rollout、rows、rewards 详情的 ValueError；否则取第一个作为 shared reward 标量。
3. 归一化与广播保持原语义：rollout_rewards 由每个 rollout 一个共享标量组成，均值减除与 GRPO/GSPO 的 std 归一化（grpo_std_normalization 时除以 std + 1e-6）逻辑不变；归一化后的值按 rollout 组广播给所有 sibling segment。
4. 测试契约更新：在 tests/fast/ray/rollout/test_train_data_conversion.py 中，把 test_grpo_uses_most_trainable_sibling_reward_for_rollout 改写为 test_grpo_rejects_different_sibling_rewards（断言 ValueError 及错误信息）；test_grpo_mainstream_count_matches_final_training_mask 改写为 test_grpo_shared_reward_ignores_final_training_mask（验证 remove_sample /

response_length 不再影响归一化）；test_grpo_mainstream_ties_use_first_sibling 改写为 test_grpo_shared_reward_uses_selected_reward_key（验证 reward_key="score" 时以选定标量做共享校验）。

5. 入口与外围不变：_post_process_rewards 仍在 grpo / gspo / reinforce_plus_plus_baseline 且 rewards_normalization 时调用本函数；custom_reward_post_process_func 优先级保持，不受影响。第二个 commit 是 merge main，用于同步主线改动。

关键文件：

- miles/ray/rollout/train_data_conversion.py（模块 奖励处理；类别 source；类型 core-logic；符号 _trainable_token_count, _normalize_rewards_by_rollout）：核心修改：删除 _trainable_token_count 启发式，_normalize_rewards_by_rollout 改为校验 sibling 奖励相等后按 rollout 广播归一化值。
- tests/fast/ray/rollout/test_train_data_conversion.py（模块 数据转换；类别 test；类型 test-coverage；符号 test_grpo_rejects_different_sibling_rewards, test_grpo_shared_reward_ignores_final_training_mask, test_grpo_shared_reward_uses_selected_reward_key）：测试从 mainstream 启发式用例改为 rejection、mask-independence、reward_key 三组新用例，验证新数据契约。

关键符号：_normalize_rewards_by_rollout, _trainable_token_count, test_grpo_rejects_different_sibling_rewards, test_grpo_shared_reward_ignores_final_training_mask, test_grpo_shared_reward_uses_selected_reward_key

关键源码片段

miles/ray/rollout/train_data_conversion.py

核心修改：删除 _trainable_token_count 启发式，_normalize_rewards_by_rollout 改为校验 sibling 奖励相等后按 rollout 广播归一化值。

```
def _normalize_rewards_by_rollout(
    args: Any,
    samples: list[Sample],
    raw_rewards: list[float],
    prompt_group_sizes: list[int] | None,
) -> list[float]:
    """每个 rollout 归一化一个共享奖励，然后把归一化结果广播给所有 sibling。"""
    if not samples:
        return []

    normalized_rewards = torch.empty(len(raw_rewards), dtype=torch.float)
    for prompt_segments in _reward_group_segments(args, samples, prompt_group_sizes):
        # 先按 rollout_id / index / row 把同一 prompt 下的样本聚成 rollout 组
        segments_by_rollout_key: dict[int | tuple[str, int], list[int]] = {}
        for segment_index in prompt_segments:
            sample = samples[segment_index]
            if sample.rollout_id is not None:
                rollout_key = sample.rollout_id
```

```
            elif sample.index is not None:
                rollout_key = sample.index
            else:
                rollout_key = ("row", segment_index)
            segments_by_rollout_key.setdefault(rollout_key, []).append(segment_index)

        rollout_segment_groups = list(segments_by_rollout_key.items())
        shared_rewards: list[float] = []
        for rollout_key, rollout_segments in rollout_segment_groups:
            # 同一个 rollout 的所有 sibling 必须共享同一奖励值；reward_key 已在
            # sample.get_reward_value() 里选定，这里做的是选定后的数据契约校验。
            sibling_rewards = [raw_rewards[segment_index] for segment_index in rollout_segments]
            if any(reward != sibling_rewards[0] for reward in sibling_rewards[1:]):
                raise ValueError(
                    f"all samples in rollout {rollout_key!r} must share one reward; "
                    f"rows {rollout_segments} have rewards {sibling_rewards}"
                )
            shared_rewards.append(sibling_rewards[0])

        # 每个 rollout 只贡献一个标量参与均值与标准差归一化，最终按组广播
        rollout_rewards = torch.tensor(shared_rewards, dtype=torch.float)
        normalized_rollout_rewards = rollout_rewards - rollout_rewards.mean()
        if args.advantage_estimator in ["grpo", "gspo"] and args.grpo_std_normalization and
        len(rollout_rewards) > 1:
            rollout_std = rollout_rewards.std()
            if rollout_std > 0:
                normalized_rollout_rewards = normalized_rollout_rewards / (rollout_std + 1e-6)

        for (_, rollout_segments), normalized_reward in zip(
            rollout_segment_groups, normalized_rollout_rewards.tolist(), strict=True
        ):
            for segment_index in rollout_segments:
                normalized_rewards[segment_index] = normalized_reward

    return normalized_rewards.tolist()
```

tests/fast/ray/rollout/test_train_data_conversion.py

测试从 mainstream 启发式用例改为 rejection、mask-independence、reward_key 三组新用例，验证新数据契约。

```
def test_grpo_rejects_different_sibling_rewards(self):
    # 同一个 rollout_id=11 下出现 2.0 与 6.0 两个奖励，旧实现会选
    # trainable token 数最多的 leaf 作为 mainstream，现在直接抛 ValueError。
    args = make_args(
        advantage_estimator="grpo",
        rewards_normalization=True,
        grpo_std_normalization=False,
        n_samples_per_prompt=2,
        rollout_batch_size=1,
```

```
    )
    samples = [
        make_sample(group_index=0, index=0, rollout_id=10, reward=0.0, loss_mask=[1, 1, 1, 1]),
        make_sample(group_index=0, index=1, rollout_id=11, reward=2.0, loss_mask=[1, 0, 0, 0]),
        make_sample(group_index=0, index=1, rollout_id=11, reward=6.0, loss_mask=[1, 1, 1, 0]),
    ]

    with pytest.raises(
        ValueError,
        match=r"all samples in rollout 11 must share one reward; rows \[1, 2\] have rewards \[2.0,
        6.0\]",
    ):
        _post_process_rewards(args, samples, custom_reward_post_process_func=None)
```

评论区精华

两位 reviewer（Shi-Dong、yueming-yuan）均给出 APPROVE，review 评论为空，Shi-Dong 仅回复 Thanks!，没有针对实现提出异议的讨论。核心设计权衡（放弃 mainstream 启发式、改为硬失败）全部沉淀在 PR body 的 Root Cause 与 Review Focus 中，且已被 reviewer 确认，无未解决疑虑。

- 暂无高价值评论线程

风险与影响

- 风险：行为从静默降级变为直接报错：若上游数据源（如 legacy fixed-fanout 路径）仍可能产出 sibling 奖励不一致的样本，训练会在 _normalize_rewards_by_rollout 处 fail-fast。PR body 说明 session v2 的 trajectory-reward broadcasting 是共享奖励的上游来源，但仍需确认所有数据源都满足该契约。影响面在 miles/ray/rollout/train_data_conversion.py 的 rollout 侧 reward 后处理与 DP 分片之间的关键路径，波及所有开启 rewards_normalization 的 GRPO/GSPO/REINFORCE++ 训练。兼容性上，_trainable_token_count 是模块私有函数，删除后若仓库其他位置引用会 ImportError，本次变更仅保留了 _normalize_rewards_by_rollout 的调用点。测试新增用例覆盖 ValueError 路径、mask 独立性、reward_key 选择，但未单独覆盖 gspo / reinforce_plus_plus_baseline 组合（与本 PR 共用代码路径，风险较低）。
- 影响：影响所有使用 rewards_normalization 的 GRPO/GSPO/REINFORCE++ 训练数据转换路径：同一 rollout 内 sibling 奖励不一致时，行为从静默选主流变为显式报错。对 session v2 用户影响较小（上游已保证共享奖励），但 legacy 固定扇出路径可能暴露数据不一致问题。团队层面，该改动强化了 reward 数据契约，避免 mask 或 leaf 长度无意间改变 advantage，提升训练统计的稳定性与可诊断性。
- 风险标记：核心训练数据路径变更，静默降级改为直接报错，依赖上游 session 奖励一致性，私有函数删除

关联脉络

- PR #2369 fix(rollout): normalize rewards per rollout: 同一文件 miles/ray/rollout/train_data_conversion.py 的上一轮 rollout 归一化实现，本 PR 收紧该归一化要求的数据契约。
- PR #2278 feat(session): request and assemble additional R3 rows under in-place weight updates: session 轨迹奖励广播链路的一部分，是共享 reward 的上游来源，与本 PR 的 sibling 共享奖励契约相关。
- PR #2368 fix(rollout): group session v2 leaf samples: session v2 多叶子样本共享 rollout_id，与 rollout 内 sibling 分组概念直接相关。