

PR #6806 完整报告

verl-project/verl

[rollout, algo] fix: compute rollout_is_seq_fraction from raw weights

合并时间: 2026-06-26 09:40

原文链接: <http://prhub.com.cn/verl-project/verl/pull/6806>

执行摘要

- 一句话: 修复 rollout_is_seq_fraction 无法跳变
- 推荐动作: 建议合并。虽然改动小但修复了一个真实死指标, 便于用户监控极端权重比例。

功能与动机

修复一个结构性死指标, TIS 场景下每个权重都被 clamp 到 $\leq \text{rollout_is_threshold}$, 导致 per-sequence 均值不可能超过阈值, 从而使 `rollout_is_seq_fraction_high` 永远为 0。详见 PR body。

实现拆解

1. 修改 `compute_is_metrics` 指标计算: 在 `verl/trainer/ppo/rollout_corr_helper.py` 中, 将原本使用 `seq_mean_weights` (clamp 后权重) 的 `rollout_is_seq_fraction_high` 和 `rollout_is_seq_fraction_low` 改为使用 `raw_seq_mean_weights` (原始权重), 通过 `verl_F.masked_mean(raw_rollout_is_weights, response_mask, axis=-1)` 得到。
2. 保留描述性指标: `rollout_is_seq_mean`、`rollout_is_seq_std`、`rollout_is_seq_max`、`rollout_is_seq_min` 等仍基于 clamp 后权重, 因为这些指标描述的是实际应用的权重。
3. 新增单元测试: 在 `tests/trainer/ppo/test_rollout_corr.py` 中新增 `test_seq_fraction_high_uses_raw_weights`, 构造一组数据覆盖原始均值超阈值而 clamp 后均值不超的情形, 断言修复后指标正确。

关键文件:

- `verl/trainer/ppo/rollout_corr_helper.py` (模块 训练器; 类别 source; 类型 core-logic; 符号 `compute_is_metrics`): 核心修复文件, 修改 `compute_is_metrics` 中的指标计算逻辑, 将 `seq_mean_weights` 替换为 `raw_seq_mean_weights`。
- `tests/trainer/ppo/test_rollout_corr.py` (模块 评分校正; 类别 test; 类型 test-coverage; 符号 `test_seq_fraction_high_uses_raw_weights`): 新增单元测试, 验证修复后的指标从原始权重计算。

关键符号: `compute_is_metrics`

关键源码片段

`verl/trainer/ppo/rollout_corr_helper.py`

核心修复文件，修改 `compute_is_metrics` 中的指标计算逻辑，将 `seq_mean_weights` 替换为 `raw_seq_mean_weights`。

```
# verl/trainer/ppo/rollout_corr_helper.py, compute_is_metrics 函数末尾
# 原代码使用 seq_mean_weights (clamp 后权重的序列均值) ,
# 导致 rollout_is_seq_fraction_high 永远为 0, 因为每个权重都被 clamp 到阈值以下。
# 修改后使用 raw_seq_mean_weights (原始权重的序列均值) ,
# 使这些指标能真正反映极端权重比例。
# 保留 seq_mean_weights 用于描述性指标 (均值 / 标准差 / 最值) ,
# 因为它们描述的是实际应用的权重。
raw_seq_mean_weights: torch.Tensor = verl_F.masked_mean(raw_rollout_is_weights, response_mask, axis=-1)
metrics["rollout_is_seq_fraction_high"] = (raw_seq_mean_weights > rollout_is_threshold).float().mean().item()
metrics["rollout_is_seq_fraction_low"] = (
    (raw_seq_mean_weights < rollout_is_threshold_lower).float().mean().item()
)
```

tests/trainer/ppo/test_rollout_corr.py

新增单元测试，验证修复后的指标从原始权重计算。

```
# tests/trainer/ppo/test_rollout_corr.py
# 新增测试函数，验证 rollout_is_seq_fraction_high 基于原始权重计算
# 设计：sequence 0 原始均值 3.5 > 2.0, clamp 后均值 1.5 < 2.0;
# sequence 1 原始和 clamp 后均值均 < 2.0。
# 若使用 clamp 后权重，high fraction 应为 0 (原 bug) ;
# 正确实现应返回 0.5 (seq0 超，seq1 不超) 。
def test_seq_fraction_high_uses_raw_weights():
    device = "cuda" if torch.cuda.is_available() else "cpu"
    threshold = 2.0
    raw_rollout_is_weights = torch.tensor([[1.0, 6.0], [0.9, 1.1]], device=device)
    rollout_is_weights = raw_rollout_is_weights.clamp(max=threshold)
    response_mask = torch.ones_like(raw_rollout_is_weights)
    log_ratio_for_metrics = torch.log(raw_rollout_is_weights)
    metrics = compute_is_metrics(
        rollout_is_weights=rollout_is_weights,
        raw_rollout_is_weights=raw_rollout_is_weights,
        log_ratio_for_metrics=log_ratio_for_metrics,
        response_mask=response_mask,
        rollout_is="token",
        rollout_is_threshold=threshold,
    )
    assert metrics["rollout_is_seq_fraction_high"] == pytest.approx(0.5)
```

评论区精华

无实质性 reviewer 讨论；代码审查机器人 `gemini-code-assist` 自动生成总结但无反馈；Luosuu 审核通过。

- 暂无高价值评论线程

风险与影响

- 风险：风险极低：变更只影响两个死指标的计算源，不影响实际权重或训练结果；新增测试覆盖该逻辑；描述性指标保持不变。
- 影响：影响范围仅限于 `compute_is_metrics` 输出的 `rollout_is_seq_fraction_high` 和 `rollout_is_seq_fraction_low` 两个指标，使之变活，对训练流程无影响。
- 风险标记：暂无

关联脉络

- 暂无明显关联 PR