PR #2765 完整报告

radixark/miles

feat(metrics): log per-rollout response lengths

合并时间：2026-08-28 05:54

原文链接：http://prhub.com.cn/radixark/miles/pull/2765

执行摘要

- 一句话：新增 per-rollout 响应长度指标，覆盖 compaction 场景
- 推荐动作：值得精读。该 PR 展示如何在指标层正确处理 session compaction 的语义（sibling 求和避免重复计数），以及如何通过统一的 episode identity 辅助函数同时修复既有 bug 并支撑新功能；review 中 P1/P2 都是真实可复现的指标错误，修复思路与测试覆盖值得借鉴。

功能与动机

Session compaction 会把一个 rollout 拆成多个训练样本，现有 per-sample 指标无法反映原始 episode 的实际训练 token 量，需要补充 per-rollout 视角；同时 Multi-LoRA 下各个 adapter 的 rollout ID 独立编号，直接聚合会产生错误的统计结果。PR body 中明确说明要记录 'per-episode response-length metrics in addition to the existing per-sample metrics'，并强调 'Shared prefixes across compacted sibling samples are counted once'。

实现拆解

1. 提取统一的 episode 身份解析：在 miles/ray/rollout/metrics.py 中新增 _get_rollout_key(sample, position)，统一处理 rollout_id、index、position 三种样本身份来源，并把 sample.adapter 纳入 key 作用域，供 _compute_training_sample_metrics 与新函数共用。这一步同时修复了 _compute_training_sample_metrics 原先在 Multi-LoRA 下缺少 adapter 作用域、可能合并不同 adapter episode 的隐患。
2. 新增 per-rollout 长度聚合函数：实现 _compute_episode_response_length_metrics(samples)。先判断是否存在 sample.adapter is not None，有则整体返回空字典（Multi-LoRA 跳过）；否则遍历样本，用 _get_rollout_key 分组，对每组累加 effective_response_length（remove_sample=True 时计为 0）得到可训练长度，再累加 sample.response_length 得到总长度；最后用 compute_statistics 输出 episode_response_length/{mean,median,max,min}，并对各 rollout 的总长度求平均输出 episode_total_response_length/mean。
3. 接入主流程：在 _compute_metrics_from_samples 中调用上述新函数，使其结果随 rollout/ 前缀写入日志与 tracking，与既有 response_len/* 指标并存。
4. 测试配套：tests/fast/ray/rollout/test_metrics.py 新增 TestEpisodeResponseLengthMetrics 类，覆盖 compaction 兄弟先求和、单样本等价性、空样本、total 长度统计、Multi-LoRA 跳过、remove_sample 零化共 6 个场景；合并最新

main 后 30 个测试全部通过。

5. 文档配套：examples/experimental/terminus-compaction/README.md 补充两组新指标的定义与 Multi-LoRA 下不输出的说明。

关键文件：

- miles/ray/rollout/metrics.py（模块 指标聚合；类别 source；类型 core-logic；符号 _get_rollout_key, _compute_episode_response_length_metrics, _compute_metrics_from_samples, _compute_training_sample_metrics）：核心逻辑文件：新增 _compute_episode_response_length_metrics，提取并改造 _get_rollout_key 纳入 adapter 作用域，并接入 _compute_metrics_from_samples；同时影响 _compute_training_sample_metrics 的 key 生成。
- tests/fast/ray/rollout/test_metrics.py（模块 指标测试；类别 test；类型 test-coverage；符号 TestEpisodeResponseLengthMetrics, test_compacted_siblings_are_summed_before_computing_statistics, test_single_sample_rollouts_match_sample_level_statistics, test_empty_samples_emit_no_episode_length_metrics）：新增 TestEpisodeResponseLengthMetrics 测试类，完整覆盖 compaction 求和、单样本等价性、空样本、total 长度、Multi-LoRA 跳过与 remove_sample 零化等关键语义。
- examples/experimental/terminus-compaction/README.md（模块 示例文档；类别 docs；类型 documentation）：补充新指标的语义定义，并明确 Multi-LoRA 场景不输出这些指标，帮助用户正确解读日志。

关键符号：_get_rollout_key, _compute_episode_response_length_metrics, _compute_metrics_from_samples, _compute_training_sample_metrics

关键源码片段

miles/ray/rollout/metrics.py

核心逻辑文件：新增 _compute_episode_response_length_metrics，提取并改造 _get_rollout_key 纳入 adapter 作用域，并接入 _compute_metrics_from_samples；同时影响 _compute_training_sample_metrics 的 key 生成。

```
def _get_rollout_key(sample: Sample, position: int) -> tuple[AdapterRef | None, str, int | None, int]:
    # Adapter 身份用于限定 rollout ID，因为每个 Multi-LoRA 数据源会独立编号，
    # 不同 adapter 可能复用相同的 group_index / index。
    if sample.rollout_id is not None:
        return (sample.adapter, "rollout", sample.group_index, sample.rollout_id)
    if sample.index is not None:
        return (sample.adapter, "sample", sample.group_index, sample.index)
    return (sample.adapter, "position", sample.group_index, position)


def _compute_episode_response_length_metrics(samples: list[Sample]) -> dict[str, float]:
    """按原始 rollout 聚合可训练与总响应 token 数。

    Session compaction 会把一个 rollout 拆成多个训练样本（sibling samples），
```

```
    它们共享同一个 rollout ID，因此需要先按 rollout 求和再计算批次级统计量，
    避免把同一 episode 的片段当成多个独立 episode 重复计数。
    可训练长度只统计 loss mask 命中的 token；总长度则统计每个样本中
    masked 与 unmasked 的全部响应 token。
    """
    # Multi-LoRA 下每个 adapter 拥有独立的 RolloutDataSource，rollout ID 仅在
    # 各自数据源内有效；直接聚合会错误合并多个 episode，因此在支持
    # adapter 级聚合前整体跳过。
    if any(sample.adapter is not None for sample in samples):
        return {}

    effective_lengths_by_rollout: dict[tuple[AdapterRef | None, str, int | None, int], int] = {}
    total_lengths_by_rollout: dict[tuple[AdapterRef | None, str, int | None, int], int] = {}

    for position, sample in enumerate(samples):
        rollout_key = _get_rollout_key(sample, position)
        # remove_sample 的样本在 convert_samples_to_train_data() 中会被清除
        # loss mask、不参与训练，因此在可训练长度中按 0 计数，但总 footprint 保留。
        effective_length = 0 if sample.remove_sample else sample.effective_response_length
        effective_lengths_by_rollout[rollout_key] = (
            effective_lengths_by_rollout.get(rollout_key, 0) + effective_length
        )
        total_lengths_by_rollout[rollout_key] = (
            total_lengths_by_rollout.get(rollout_key, 0) + sample.response_length
        )

    if not effective_lengths_by_rollout:
        return {}

    log_dict = {}
    log_dict |= dict_add_prefix(
        compute_statistics(list(effective_lengths_by_rollout.values())),
        "episode_response_length/",
    )
    log_dict["episode_total_response_length/mean"] = float(
        np.mean(list(total_lengths_by_rollout.values()))
    )
    return log_dict
```

tests/fast/ray/rollout/test_metrics.py

新增 TestEpisodeResponseLengthMetrics 测试类，完整覆盖 compaction 求和、单样本等价性、空样本、total 长度、Multi-LoRA 跳过与 remove_sample 零化等关键语义。

```
class TestEpisodeResponseLengthMetrics:
    def test_compacted_siblings_are_summed_before_computing_statistics(self):
        # 同一 rollout_id=10 下有两个 sibling samples（group_index=0），
        # 它们应被合并为一个 episode：可训练长度为 2 + 3 = 5，
        # 而 group_index=1 的另一个 rollout_id=10 是独立 episode（长度 6）。
        samples = [
```

```
            make_sample(group_index=0, index=0, rollout_id=10, response_length=5, loss_mask=[1,
            1, 0, 0, 0]),
            make_sample(group_index=0, index=0, rollout_id=10, response_length=7, loss_mask=[1,
            1, 1, 0, 0, 0, 0]),
            make_sample(group_index=0, index=1, rollout_id=11, response_length=4, loss_mask=[1,
            1, 1, 1]),
            make_sample(group_index=1, index=2, rollout_id=10, response_length=8, loss_mask=[1,
            1, 1, 1, 1, 1, 0, 0]),
        ]

        out = _compute_episode_response_length_metrics(samples)

        assert out == {
            "episode_response_length/mean": pytest.approx(5.0),
            "episode_response_length/median": pytest.approx(5.0),
            "episode_response_length/max": pytest.approx(6.0),
            "episode_response_length/min": pytest.approx(4.0),
            "episode_total_response_length/mean": pytest.approx(8.0),
        }

    def test_removed_sample_has_zero_effective_length_but_keeps_total_length(self):
        # remove_sample=True 的样本不参与训练：episode 可训练长度计为 0，
        # 但总长度仍保留完整 footprint，避免低估上下文占用。
        sample = make_sample(
            index=0,
            rollout_id=10,
            response_length=5,
            loss_mask=[1, 1, 1, 1, 1],
            remove_sample=True,
        )

        out = _compute_episode_response_length_metrics([sample])

        assert out["episode_response_length/mean"] == pytest.approx(0.0)
        assert out["episode_total_response_length/mean"] == pytest.approx(5.0)
```

评论区精华

review 中 guapisolo 提出两条关键问题：

- [P1] Multi-LoRA 下 key 冲突：每个 adapter 拥有独立的 RolloutDataSource，不同 adapter 会复用相同的 group_index/index，导致不同 episode 被错误合并。建议“Skip per-rollout length metrics for Multi-LoRA”，并注明 README。最终采纳为 if any(sample.adapter is not None for sample in samples): return {}。
- [P2] remove_sample 虚报可训练长度：指标在 convert_samples_to_train_data() 清除 loss mask 之前记录，会导致 dashboard 上报实际不参与训练的 token。建议“effective length 计 0、total 保持不变”，最终通过 effective_length = 0 if sample.remove_sample else sample.effective_response_length 实现。

  - claude[bot] 额外指出 _compute_training_sample_metrics 原本缺少 adapter 作用域（pre-existing），通过共享带 adapter 的 _get_rollout_key 一并修复，并有测试覆盖。最终 guapisolo 与 maocheng23 均 approve，claude[bot] 评论确认修复与测试到位。
- Multi-LoRA 下 rollout key 冲突与指标跳过 (correctness): 采纳建议：当任意 sample 带有 adapter 时整体跳过新指标，并在 README 中注明 Multi-LoRA 不输出这些指标。
- remove_sample 样本的可训练长度虚报 (correctness): 采纳建议：remove_sample 样本的 effective_length 计 0，而 total_length 保持不变。
- _compute_training_sample_metrics 缺少 adapter 作用域（pre-existing） (correctness): 通过共享带 adapter 字段的 _get_rollout_key 修复，并新增 test_rollout_ids_are_scoped_by_adapter 测试覆盖。

风险与影响

- 风险：
  1. 指标口径变化：episode_response_length/{mean,median,max,min} 是 per-rollout 聚合值，与既有 per-sample 的 response_len/* 语义不同；依赖旧口径的自动告警或看板可能产生误读。
  2. Multi-LoRA 可观测性缺口：新指标对 Multi-LoRA 整体跳过，训练不受影响，但该场景下暂时无法从 episode 维度观测 token 长度，需后续实现 adapter-scoped 聚合。
  3. remove_sample 口径差异：同一被过滤样本在 episode_response_length 中计入 0、在 episode_total_response_length 中保留完整长度，直接对比两组指标可能造成困惑，需确保文档说明清晰。
  4. 核心指标模块变更：miles/ray/rollout/metrics.py 是 rollout 可观测性的核心模块，_get_rollout_key 被 _compute_training_sample_metrics 复用，影响所有 rollout 日志计算；合并时与 recent #2778 的 adapter-scoping 改动交织，需回归 Multi-LoRA 奖励指标。 - 影响：对用户与工程师：日志和 dashboard 新增两组 rollout 维度指标，可更准确观察 compaction 场景下每个原始 episode 的实际训练 token 量与总响应 footprint，便于诊断训练吞吐与截断问题。对系统：纯统计逻辑、单次遍历，无运行时性能影响，日志键为新增、向后兼容。对团队：metrics.py 近期多个 PR（#2710、#2743、#2778）持续演进，本次提取 _get_rollout_key 形成统一的 episode 身份解析，为后续 adapter-scoped 聚合与更多 episode 级指标打了基础。 - 风险标记：Multi-LoRA 指标缺口，指标口径变化，核心指标模块变更

关联脉络

- PR #2778 fix(metrics): scope episode rewards by adapter: 本 PR body 明确引用 #2778：episode rewards 保持 adapter-scoped；测试中复用 #2778 的 adapter-scoping 测试，且共享同一事件身份解析逻辑。
- PR #2741 Add Terminus 2 compaction training example: 本 PR 修改了该示例的 README.md，补充新指标的文档说明，属于同一 compaction 功能线。
- PR #2710 Log compaction-aware rollout metrics: 同文件 metrics.py 中 compaction 相关指标的早期演进，本次是 per-rollout 长度指标的延续。

- PR #2743 Skip non-numeric rewards in the episode average: 同样修改 miles/ray/rollout/metrics.py，体现该模块指标健壮性的一系列修复。