

PR #2817 完整报告

radixark/miles

dashboard: give a step with no samples the summary schema

合并时间: 2026-09-01 04:32

原文链接: <http://prhub.com.cn/radixark/miles/pull/2817>

执行摘要

- 一句话: 空 step 补全声明式 schema, 修复 dashboard 多页面 500 崩溃
- 推荐动作: 值得精读。改动量不大但信息密度高: polars 空数据 schema 推断陷阱、缓存版本在格式迁移中的作用 (mtime 匹配 $\neq$ 格式有效)、并发读下的原子写、以及「先用 revert 实验证明测试能抓到 bug 再提交」的验证方法 (PR body 中 2 failed 的对照表)。对 dashboard 维护者、做数据管道容错的工程师尤其有参考价值; blank_samples 夹具和 test_pre_fix_columnless_cache_is_not_served 是两段值得模仿的测试写法。

功能与动机

PR body 明确定位: #2815 只是停止让 Rollouts 页「落在」不可用 step 上, 底层崩溃仍在。一个 step 可能在生成落地前被 abort, dump 中没有任何 sample; `summary()` 从空 row list 推断 schema, 得到的 frame 是零列而非零行, 导致每个消费方崩溃: `groups()` 的 `ColumnNotFoundError: unable to find column "group_index"` 没有对应 handler (rollout 页 500)、trajectories endpoint 索引 `summary()["sample_index"]` 同样崩溃、`step_aggregates()` 遍历每个 step, 一个空 step 拖垮整个 dump 派生 metrics 页。作者因此强调要在 reader 层修复而不是 patch `groups()`。

实现拆解

1. 空 schema 兜底 (核心): miles/dashboard/dump_reader.py 的 DumpReader 新增类常量 SUMMARY_COLUMNS (30 列元组, 顺序与 _summary_row 输出一致)。summary() 将列表推导结果存入 rows, 为空时改用 `pl.DataFrame(schema={name: pl.Null for name in self.SUMMARY_COLUMNS})` 构造 0 行 30 列全 Null 帧。三个下游消费者据此读到空表: `groups()` 返回空分组、trajectories endpoint 索引 sample_index 得空列表、`step_aggregates()` 如实上报 n_samples: 0 且 aggregate 为 None, 邻居 step 不受影响。dtype 故意不钉死: spec_accept_rate 在全 None 时推断为 Null、有值时是 Float64, 空帧的 Null 可在下游聚合中干净转换。
2. 缓存一致性 (版本迁移 + 原子写): 仅改 summary() 不够——修复前的 columnless parquet 的 sources.json (文件 mtime + _summary_version: 4) 与 dump 现状完全匹配, 快速路径会继续返回坏缓存, 因此 SUMMARY_VERSION 提升到 5。同时缓存写入改为「先写 `cache_path.with_suffix(f".{os.getpid()}.tmp")` 再 replace»: dashboard handler 是 sync def, FastAPI 在线程池运行, 一次页面加载并发触发 /summary、/groups、/trajectories、step_aggregates(), 直写会让并发读者拿到被截断的 parquet 抛

ComputeError (`_translate_errors` 不映射, 500) 。

3. 列顺序统一: `_summary_row` 的 `train-absent` 分支原用 `dict.fromkeys` 构造 `train` 列再后补 `alignment_failed`、`truncated`, 顺序与 `train-present` 分支不同 (`truncated` 排最后)。`polars` 从 `list-of-dict` 的第一行决定列顺序, `eval dump` 的 `sample` 全走 `train-absent` 分支, 导致 `summary(id, evaluation=True).columns` 与 `SUMMARY_COLUMNS` 永久错位。改为显式字面量后两分支列序一致。
4. `advisories` 防护: `miles/dashboard/advisory.py` 的 `_rollout_advisories` 检查的「最新 `train step`」恰是 `abort` 时最可能为空的那个。空 `step` 带完整 `schema` 后 "`alignment_failed`" in `summary.columns` 从 `False` 翻转为 `True`, 全 `Null` 列的 `Series.any()` 抛 `SchemaError` (`_translate_errors` 不映射, `/api/advisory 500`)。加 if `summary.height == 0`: `return out` 提前返回。
5. 测试配套: `tests/fast/dashboard/dummy_dump.py` 新增 `blank_samples()` 夹具, 忠实模拟 `abort` 后 `writer` 的全部残留 (空 `samples`、空 `dashboard_columns` `parquet`、无 `trajectory sidecar`、无 `train shards`、`mtime` 回拨)。`test_dump_reader_views.py` 新增 4 个测试: 钉住 `SUMMARY_COLUMNS` 与真实 `summary` 列一致 (`train` 与 `eval` 两方向); 空 `step` 置于 3-step `dump` 中间, 强制 `step_aggregates()` 走过并断言邻居完好; 全 `Null` 帧经 `parquet` 缓存往返不丢 `schema`; 手工构造 `_summary_version=4` 旧缓存确认被丢弃。`test_advisory.py` 新增空 `step` 下 `compute_advisories` 不抛异常的测试。

关键文件:

- `miles/dashboard/dump_reader.py` (模块 读取器; 类别 `source`; 类型 `core-logic`; 符号 `DumpReader.summary`, `DumpReader._summary_row`, `DumpReader.SUMMARY_COLUMNS`, `DumpReader.SUMMARY_VERSION`): 核心修复文件: `summary()` 在 `rows` 为空时用 `SUMMARY_COLUMNS` 构造 0 行全 `Null` 帧; `SUMMARY_VERSION 4 → 5` 使旧 `columnless` 缓存失效; `parquet` 缓存改原子写; `_summary_row` 两个分支列顺序统一。
- `tests/fast/dashboard/test_dump_reader_views.py` (模块 视图测试; 类别 `test`; 类型 `test-coverage`; 符号 `test_summary_columns_declaration_matches_reality`, `test_step_with_no_samples_reads_as_empty_not_broken`, `test_empty_step_survives_the_parquet_cache`, `test_pre_fix_columnless_cache_is_not_served`): 新增 4 个测试: 钉住 `SUMMARY_COLUMNS` 防漂移、中间空 `step` 的行为验证、`parquet` 缓存往返、修复前旧缓存不被服务; 空 `step` 刻意放中间以证明 `step_aggregates` 遍历时邻居完好。
- `tests/fast/dashboard/dummy_dump.py` (模块 测试夹具; 类别 `test`; 类型 `test-coverage`; 符号 `blank_samples`): 新增 `blank_samples` 夹具, 忠实模拟 `abort` 后 `writer` 留下的全部产物 (空 `samples`、空 `dashboard_columns`、无 `trajectory`、无 `train shards`), 是所有空 `step` 测试的基础。
- `miles/dashboard/advisory.py` (模块 告警逻辑; 类别 `source`; 类型 `core-logic`; 符号 `_rollout_advisories`): 连带修复: `_rollout_advisories` 对空 `step` 提前返回, 避免空 `step` 带完整 `schema` 后 `alignment_failed` 全 `Null` 列触发 `SchemaError` 导致 `/api/advisory 500`。

- tests/fast/dashboard/test_advisory.py (模块 告警测试; 类别 test; 类型 test-coverage ; 符号 test_blank_newest_step_does_not_break_rollout_advisories) : 新增 test_blank_newest_step_does_not_break_rollout_advisories, 覆盖 compute_advisories 与空 step reader 的组合, 防止 /api/advisory 回归 500。

关键符号: DumpReader.summary, DumpReader._summary_row, DumpReader.SUMMARY_COLUMNS, DumpReader.SUMMARY_VERSION, _rollout_advisories, blank_samples

关键源码片段

miles/dashboard/dump_reader.py

核心修复文件: summary() 在 rows 为空时用 SUMMARY_COLUMNS 构造 0 行全 Null 帧; SUMMARY_VERSION 4 → 5 使旧 columnless 缓存失效; parquet 缓存改原子写; _summary_row 两个分支列顺序统一。

```
def summary(self, rollout_id: int, *, evaluation: bool = False) -> pl.DataFrame:
    """Per-sample summary table (每个 Sample 一行), parquet 缓存于 cache_dir,
    以源文件 mtime 与 SUMMARY_VERSION 共同作为失效依据。"""
    stem = f"rollout_{'eval_' if evaluation else ''}{rollout_id}"
    cache_path = self.cache_dir / f"{stem}.parquet"
    sources_path = self.cache_dir / f"{stem}.sources.json"
    sources = self._source_stamps(rollout_id, evaluation=evaluation)
    # 快速路径: parquet 与 sources.json 都命中且 mtime 戳一致才直接读缓存。
    # SUMMARY_VERSION 参与比对, 保证 schema 变更后旧缓存不被继续服务。
    if cache_path.exists() and sources_path.exists() and json.loads(sources_path.read_text()) ==
        sources:
        return pl.read_parquet(cache_path)

    joined = self.joined(rollout_id, evaluation=evaluation)
    rows = [self._summary_row(s, joined.train_rows.get(s.index), rollout_id=rollout_id) for s in
        joined.samples]
    # 空 rows 时 polars 会推断出零列 frame, 下游 groups()/step_aggregates()
    # 全部因 ColumnNotFoundError 崩溃。这里显式用声明式 schema 构造
    # 0 行 30 列的全 Null 帧, 让消费者读到「空 step」而不是「坏 frame」。
    # SUMMARY_COLUMNS 列序与 _summary_row 输出一致, 由测试钉住防漂移。
    df = (
        pl.DataFrame(rows, strict=False)
        if rows
        else pl.DataFrame(schema={name: pl.Null for name in self.SUMMARY_COLUMNS})
    )
    self.cache_dir.mkdir(parents=True, exist_ok=True)
    # 原子写: handler 是 sync def, FastAPI 把它丢进线程池, 同一页面会并发
    # 触发 /summary、/groups、/trajectories 等读取; 直写最终路径会让并发
    # 读者拿到被截断的 parquet 而抛 ComputeError (_translate_errors 不映射, 500)。
    tmp_path = cache_path.with_suffix(f".{os.getpid()}.tmp")
    df.write_parquet(tmp_path)
    tmp_path.replace(cache_path)
```

```
sources_path.write_text(json.dumps(sources))
return df
```

tests/fast/dashboard/dummy_dump.py

新增 blank_samples 夹具，忠实模拟 abort 后 writer 留下的全部产物（空 samples、空 dashboard_columns、无 trajectory、无 train shards），是所有空 step 测试的基础。

```
def blank_samples(dump_dir: Path, rollout_id: int, *, seconds: float = 100.0) -> None:
    """把某个 step 重写为「生成落地前 abort」的状态，用于构造空 step 测试场景。"""
    path = dump_dir / "rollout_data" / f"{rollout_id}.pt"
    pack = torch.load(path, map_location="cpu", weights_only=False)
    pack["samples"] = []
    torch.save(pack, path)

    # 真实 abort 后 writer 会留下空但带列的 dashboard_columns parquet,
    # 且不会写 trajectory sidecar; 若不同步清理, /tokens 端点会对 dump
    # 中已不存在的 sample 继续作答。
    save_dashboard_columns([], dump_dir / "dashboard_columns" / f"rollout_{rollout_id}.parquet")
    (dump_dir / "trajectory" / f"{rollout_id}.jsonl").unlink(missing_ok=True)
    for shard in (dump_dir / "train_data").glob(f"{rollout_id}_*.pt"):
        shard.unlink()

    # 回拨 mtime, 让文件通过 DumpReader 的可见性守卫 (MIN_AGE_SECONDS)
    stamp = time.time() - seconds
    os.utime(path, (stamp, stamp))
```

miles/dashboard/advisory.py

连带修复：_rollout_advisories 对空 step 提前返回，避免空 step 带完整 schema 后 alignment_failed 全 Null 列触发 SchemaError 导致 /api/advisory 500。

```
def _rollout_advisories(reader: DumpReader, args: dict) -> list[Advisory]:
    """对最新完成的 train rollout 做退化信号检查。"""
    train_ids = reader.rollout_ids().train
    if not train_ids:
        return []
    rollout_id = train_ids[-1]
    out = []

    groups = reader.groups(rollout_id)
    with_rewards = groups.filter(groups["reward_mean"].is_not_null())
    if len(with_rewards):
        zero_std = int(with_rewards["zero_std"].sum())
        if zero_std / len(with_rewards) >= ZERO_STD_FRAC_WARN:
            message = (
                f"{zero_std}/{len(with_rewards)} groups in rollout {rollout_id} "
                "have zero reward std — those groups contribute zero policy gradient"
            )
            if with_rewards["reward_mean"].abs().max() < 1e-12:
                message += (
```

```

        "; every reward is 0, pointing at systematic failure "
        "(timeouts/truncation), not task difficulty"
    )
    out.append(Advisory(level="warning", message=message))

summary = reader.summary(rollout_id)
# 空 step 修复后 summary 带完整 schema, alignment_failed 列全为 Null:
# "alignment_failed in summary.columns" 从 False 翻转为 True, 而 Null
# dtype 列上的 Series.any() 会抛 SchemaError (_translate_errors 不映射,
# /api/advisory 会 500)。空 step 没有可诊断信号, 提前返回。
if summary.height == 0:
    return out

truncated_frac = summary["truncated"].cast(float).mean()
if truncated_frac is not None and truncated_frac >= TRUNCATED_FRAC_WARN:
    message = f"{truncated_frac:.0%} of samples in rollout {rollout_id} are truncated"
    cap = args.get("rollout_max_response_len")
    if cap:
        message += (
            f" — check --rollout-max-response-len ({cap:g}): in multi-turn agentic runs "
            "the per-turn cap is usually the binding limit, not the context length"
        )
    out.append(Advisory(level="warning", message=message))

if "alignment_failed" in summary.columns and bool(summary["alignment_failed"].any()):
    unreadable = int(summary["alignment_failed"].sum())
    out.append(
        Advisory(
            level="info",
            message=(
                f"per-token columns are blank for {unreadable} samples in rollout {rollout_id}: "
                "their context-parallel slices could not be placed, because this dump predates "
                "the cp_rank/cp_size fields. Blank here means unreadable, not undumped"
            ),
        ),
    )
)
return out

```

评论区精华

本 PR 的 review 主要由 claude[bot] 完成，核心交锋在第一轮：[SUMMARY_VERSION](#) 未提升会让修复前的 columnless parquet 继续被快速路径服务（sources.json 的 mtime 与版本比对全部通过），修复等于没生效。该问题通过后续 commit [065890d](#) 解决，并配套 [test_pre_fix_columnless_cache_is_not_served](#)。第二轮 claude[bot] 确认「Nothing blocking」，并核对 [SUMMARY_COLUMNS](#) 声明顺序与 [_summary_row](#) 两个分支 dict 键顺序的差异，结论是 polars 按键名匹配值不会错位，但列顺序会与声明不一致（由 Zhichenzzz 的 [4a33fe7](#) 提交处理）。另外 Zhichenzzz 作为协作者在过程中补了三个连带修复：原子写、列序统一、advisories 空 step 防护，都是 review 之外的深入挖掘。

- SUMMARY_VERSION 未提升，修复前的 columnless 缓存会被快速路径继续服务 (correctness): commit 065890d 将 SUMMARY_VERSION 从 4 提升到 5，并新增 test_pre_fix_columnless_cache_is_not_served 构造 v4 旧缓存验证被丢弃；claude[bot] 下轮确认修复到位。
- SUMMARY_COLUMNS 声明顺序与 _summary_row 两个分支的 dict 键顺序差异 (design): 统一两个分支的 dict 顺序 (truncated 紧跟 normalized_reward)，并让 test_summary_columns_declaration_matches_reality 同时钉住 train 与 eval 两个方向。

风险与影响

- 风险：缓存迁移：SUMMARY_VERSION 提升会让线上已存在的 dashboard 缓存目录在首次访问时全部重建，代价是一次性重算所有 step 的 summary；若未来修改列集合而忘记提升版本，test_pre_fix_columnless_cache_is_not_served 只钉住了这一次迁移，其他迁移需靠 test_summary_columns_declaration_matches_reality 防漂移。原子写残留：write_parquet 与 replace 之间进程崩溃会遗留 {pid}.tmp 文件，不影响正确性但会积累垃圾。行为语义变化：空 step 从「整页 500」变为「空表格 + n_samples: 0 + null aggregates」，advisories 对空 step 不再输出 truncated/alignment 警告，这是有意的，但依赖 summary.height == 0 守卫，未来若给空帧追加非 Null 列需重新审视。测试覆盖边界：blank_samples 已尽力复刻 abort 残留物，但真实 dump 变体（部分 rank 文件缺失、dashboard_columns 维数异常）仍可能有未覆盖组合。
- 影响：用户（训练 / 评估团队）：dashboard 的 Rollouts 页、metrics 页、trajectories 接口、advisory 接口在运行中出现空 step（如训练在生成落地前 abort）时不再返回 500，而是渲染空表格并如实上报 n_samples: 0；一次中止不再毒化整页指标。系统：dashboard 缓存目录会在部署后首次访问时全部重建（SUMMARY_VERSION 提升）；并发读缓存窗口被原子写关闭，消除了 ComputeError 500。团队：测试基建新增 blank_samples 夹具，后续空 step 相关测试可复用；PR 展示了「claude[bot] review 发现问题 → 协作者补 commit + 回归测试」的闭环协作模式。
- 风险标记：缓存格式变更需版本迁移，空数据路径影响多个 endpoint，常量与实现双源需测试钉住，并发读缓存需原子写

关联脉络

- PR #2815（标题未提供；PR body 称其阻止 Rollouts 页落在不可用 step 上）：PR body 明确说明本 PR 是 #2815 的 follow-up：#2815 只让 Rollouts 标签页不落在不可用 step 上，未修复底层崩溃，本 PR 在 reader 层根治。
- PR #2794 dashboard: scroll the token strip instead of paging through it: 同一 dashboard 模块的交互与性能优化，与 #2817 共同体现 dashboard 的持续打磨。
- PR #2743 Skip non-numeric rewards in the episode average: 同类「异常数据下 dashboard/observability 不崩溃」的健壮性修复：跳过非数值奖励避免训练崩溃，本 PR 让空 step 不再拖垮指标页。