PR #2278 完整报告

radixark/miles

feat(session): request and assemble additional R3 rows under in-place weight updates

合并时间：2026-08-12 13:43

原文链接：http://prhub.com.cn/radixark/miles/pull/2278

执行摘要

- 一句话：in_place 模式下按增量请求并拼接 R3，削减多轮重复传输
- 推荐动作：值得精读。核心设计是『把 R3 看作严格 append-only patch 流』：checkpoint 长度作为严格边界、普通 merge 决定物化前缀、LCP 仅作 sanity assertion，三者职责清晰；测试金字塔完整（e2e oracle 逐字节比对 + fast 单测覆盖分支 / 截断 / 异常），对做长会话 RL 训练或 session 服务化的同学有直接参考价值。

功能与动机

TITO session 每轮都重发完整 token 序列，配合 routing replay 时 SGLang 会对整个前缀重新返回 routed-expert 行，累积 R3 payload 随所有轮次长度之和增长。PR body 原话：『SGLang returns routed-expert rows for the whole prefix again and the cumulative R3 payload grows with the sum of all turn lengths.』本 PR 让 SessionServer 通过 routed_experts_start_len 只请求 checkpoint 之后新增的行，并在 samples 组装时按 append-only 语义还原完整张量，既降低传输与计算开销，又保持训练侧张量与 full-R3 完全一致。

实现拆解

实现按以下 5 步拆解：

1. 能力推导与参数校验：miles/rollout/session/server.py 在 SessionServer 启动时从 pause_generation_mode 推导内部能力标志 use_addition_r3（in_place 为 True，retract 与缺省为 False），并通过 sessions.py/v2/utils.py 透传给 SessionCore/SessionCoreV2；miles/utils/arguments.py 新增校验，--use-session-server 与 --pause-generation-mode=abort 组合直接断言失败，避免不兼容模式静默运行。
2. 请求侧增量偏移：miles/rollout/session/core.py 新增 _lcp_len 辅助函数与 _maybe_request_addition_r3。chat_completions 在 prepare_pretokenized 完成（即 rollback 已应用）后调用它，以 checkpoint token 数减一作为 routed_experts_start_len 写入请求体，并用 LCP 仅做稳定性断言。v2 路径在 miles/rollout/session/v2/core.py 中使用 session.active_token_ids() 作为 checkpoint。
3. 组装侧 patch 拼接：miles/rollout/session/samples/merge.py 新增 merge_samples_with_addition_r3：先走普通 merge_samples 得到最终 token 前缀，再

按 record 顺序严格拼接 patch，校验每段 start == covered_rows、payload 存在性与行数匹配，最后截断到 len(tokens) - 1 行。compute_samples_from_openai_records 增加 use_addition_r3 开关，per-turn sample 不再预先解码 R3。

4. v2 逐叶组装：SessionCoreV2.collect_samples 将标志传入 build_leaf_material，每个 root-to-leaf 路径独立物化自己的前缀 patch，支持分支树场景。
5. 测试与配套：新增 tests/e2e/sglang/test_session_server_addition_r3.py（Qwen3-30B-A3B、2×H200、十轮 TITO 与 full-R3 oracle 逐字节比对）；fast 套件覆盖 v1/v2 samples op、严格组装、截断边界、gap 422、rollback/ 分支偏移与参数互斥校验。

关键文件：

- miles/rollout/session/core.py（模块 会话核心；类别 source；类型 core-logic；符号 _lcp_len, init, _maybe_request_addition_r3）：核心逻辑入口：新增 use_addition_r3 推导标志、_maybe_request_addition_r3 请求侧偏移注入、collect_samples 分支选择 addition 组装。
- miles/rollout/session/samples/merge.py（模块 样本组装；类别 source；类型 dependency-wiring；符号 merge_samples_with_addition_r3）：新增 merge_samples_with_addition_r3，是增量 patch 拼接为完整张量的核心实现，严格校验 append 连续性。
- tests/e2e/sglang/test_session_server_addition_r3.py（模块 e2e 测试；类别 test；类型 test-coverage；符号 sglang_server, _serve_session, _decode_r3, test_tito_session_addition_r3_matches_full_r3）：端到端验证：10 轮真实 TITO 会话与 full-R3 oracle 逐字节比对，证明增量 patch 与训练张量完全等价。
- miles/rollout/session/v2/core.py（模块 会话 V2；类别 source；类型 core-logic；符号 init）：v2 树形轨迹链路透传 use_addition_r3，chat_completions 使用 active_token_ids() 作为 checkpoint，collect_samples 逐叶物化。
- tests/fast/rollout/session/test_samples.py（模块 组装单测；类别 test；类型 test-coverage；符号 _r3_rows, _r3_patch, _merge_addition, TestAdditionR3Assembly）：TestAdditionR3Assembly 覆盖严格 patch 组装：双 patch 重建全量参考、重叠 / 空洞 / 缺失 payload 拒绝、截断边界与空 delta 形状保持。
- miles/rollout/generate_utils/generate_endpoint_utils.py（模块 生成工具；类别 source；类型 core-logic；符号 get_routed_experts_from_response）：get_routed_experts_from_response 的 decoder 长度参数改名为 num_tokens，与张量轴语义对齐，被 addition 组装路径调用。

关键符号：_lcp_len, _maybe_request_addition_r3, merge_samples_with_addition_r3, compute_samples_from_openai_records, get_routed_experts_from_response

关键源码片段

miles/rollout/session/core.py

核心逻辑入口：新增 use_addition_r3 推导标志、_maybe_request_addition_r3 请求侧偏移注入、collect_samples 分支选择 addition 组装。

```
def _lcp_len(a: list[int], b: list[int]) -> int:
```

```
    """Length of the longest common prefix of two token-ID lists."""
    n = min(len(a), len(b))
    for i in range(n):
        if a[i] != b[i]:
            return i
    return n


class SessionCore:
    def __init__(
        self, backend, registry: SessionRegistry, args, session_server_instance_id=None, *, use_
        addition_r3=False
    ):
        self.backend = backend
        self.registry = registry
        self.args = args
        self.instance_id = session_server_instance_id
        # 由 pause_generation_mode 在 server 启动时推导；session 代码只依赖
        # 这个能力标志，绝不直接依赖权重更新模式本身。
        self.use_addition_r3 = use_addition_r3

    def _maybe_request_addition_r3(
        self, request_body: dict, checkpoint_token_ids: list[int], prompt_token_ids: list[int]
    ) -> None:
        """Ask SGLang to return only the R3 rows the session has not retained.

        为什么 checkpoints 的 N - 1 行必须是新 prompt 的因果前缀：
        只有前缀稳定，每个持久化 patch 才能恰好接在上一个 patch 之后，
        拼接时才能保证无重叠、无空洞。
        """
        # 仅在 addition 模式且本次请求确实开启了 routing replay 时生效；
        # retract / 全量 R3 路径完全不受影响。
        if not (self.use_addition_r3 and request_body.get("return_routed_experts")):
            return
        previous_rows = max(0, len(checkpoint_token_ids) - 1)
        stable_prefix_tokens = _lcp_len(checkpoint_token_ids, prompt_token_ids)
        assert (
            stable_prefix_tokens >= previous_rows
        ), f"additional R3 requires {previous_rows} stable prefix tokens, got {stable_prefix_tokens}"
        # SGLang 据此只返回行 [routed_experts_start_len, len(prompt + output) - 1)，
        # 把多轮累积的 R3 payload 从“所有轮次长度之和”降为“本轮回合新增行数”。
        request_body["routed_experts_start_len"] = previous_rows
```

miles/rollout/session/samples/merge.py

新增 merge_samples_with_addition_r3，是增量 patch 拼接为完整张量的核心实现，严格校验 append 连续性。

```
def merge_samples_with_addition_r3(
    args: Namespace,
```

```
    samples: list[Sample],
    records: list[SessionRecord],
    tokenizer,
) -> Sample:
    """先走普通 merge，再按 append-only 语义物化所需 R3 前缀。"""
    merged = merge_samples(samples, tokenizer)
    # 没有任何 record 带 R3（replay 未开启）时保持原行为，addition 模式休眠。
    if all(record.response["choices"][0]["meta_info"].get("routed_experts") is None for record in
    records):
        return merged

    required_rows = len(merged.tokens) - 1 # 普通 merge/ 截断已决定最终 token 前缀
    covered_rows = 0
    chunks: list[np.ndarray] = []
    for i, record in enumerate(records):
        if chunks and covered_rows >= required_rows:
            break # 已覆盖所需前缀，后续 patch 不需要再解码

        choice = record.response["choices"][0]
        info = choice["meta_info"].get("routed_experts")
        if info is None:
            raise ValueError(f"additional R3: record {i} has no routed_experts payload")

        start = record.request.get("routed_experts_start_len")
        if start is None:
            raise ValueError(f"additional R3: record {i} request carries no routed_experts_start_
            len")
        if start != covered_rows:
            # 严格 append 约束：每个 patch 必须从当前已覆盖行开始，空洞直接拒绝
            raise ValueError(f"additional R3: record {i} starts at row {start}; expected {covered_
            rows}")

        end = len(record.request["input_ids"]) + len(choice["meta_info"]["output_token_logprobs"]) -
        1
        if end < start:
            raise ValueError(f"additional R3: record {i} has invalid offsets (start={start}, end={end})
            ")
        delta_rows = end - start
        if bool(info) != bool(delta_rows):
            raise ValueError(f"additional R3: record {i} payload presence does not match {delta_
            rows} rows")

        patch = get_routed_experts_from_response(args, choice, delta_rows)
        if len(patch) or required_rows == 0:
            chunks.append(patch)
        covered_rows = end

    if covered_rows < required_rows:
        raise ValueError(
```

```
            f"additional R3 covers {covered_rows} rows but the merged sample needs "
            f"{required_rows} (len(tokens) - 1)"
        )
    # 只物化最终选定前缀对应的行，避免把截断丢弃的行也解码出来
    merged.rollout_routed_experts = np.concatenate(chunks)[:required_rows]
    return merged
```

tests/e2e/sglang/test_session_server_addition_r3.py

端到端验证：10 轮真实 TITO 会话与 full-R3 oracle 逐字节比对，证明增量 patch 与训练张量完全等价。

```
def test_tito_session_addition_r3_matches_full_r3(sglang_server):
    with _serve_session(sglang_server.base_url) as session_url:
        # ... 十轮 chat + 每轮 checkpoint 快照 ...
        # 关键设计：用"one-token oracle"生成最终 token 的完整 R3 参考，
        # 该 oracle 必须精确复现 final_token_ids[-1]，否则后续字节比对无意义。
        oracle_meta = oracle_response.json()["meta_info"]
        oracle_output_ids = [item[1] for item in oracle_meta["output_token_logprobs"]]
        assert oracle_output_ids == [final_token_ids[-1]]
        full_r3 = _decode_r3(oracle_meta["routed_experts"])
        assert len(full_r3) == (len(final_token_ids) - 1) * _ROW_BYTES

        covered_rows = 0
        for index, (record, checkpoint) in enumerate(zip(records, checkpoints, strict=True)):
            choice = record["response"]["choices"][0]
            start = record["request"]["routed_experts_start_len"]
            end = len(record["request"]["input_ids"]) + len(choice["meta_info"]["output_token_
            logprobs"]) - 1
            # 每一轮的 start 必须严格等于上一轮 end，即 append-only 流
            assert start == covered_rows
            if index:
                assert start == len(checkpoints[index - 1]) - 1
            assert end == len(checkpoint) - 1

            patch = _decode_r3(choice["meta_info"]["routed_experts"])
            # 与同轨迹的 full-R3 oracle 逐字节比对，验证增量 patch 语义正确
            assert len(patch) == (end - start) * _ROW_BYTES
            assert patch == full_r3[start * _ROW_BYTES : end * _ROW_BYTES]
            covered_rows = end

        assert covered_rows == len(final_token_ids) - 1
        # /samples 返回的最终张量也必须与 oracle 完全一致
        assert sample.rollout_routed_experts.tobytes(order="C") == full_r3
```

评论区精华

PR 无任何 review 评论，仅有一条 Shi-Dong 的 LGTM approve，核心设计讨论沉淀在 PR body 的 Design Notes 与 Review Focus 中：

- append 边界选择：_maybe_request_addition_r3 用 checkpoint 长度作为严格 append 边界，LCP 仅作 sanity assertion，确保每个持久化 patch 恰好接在上一个 patch 之后。
- 物化前缀由普通 merge 决定：merge_samples_with_addition_r3 先跑完整 merge/ 截断，再只拼接选定前缀所需的 patch，避免解码被截断丢弃的行。
- 能力不暴露为独立选项：use_addition_r3 由 pause_generation_mode 在启动时推导，不新增用户可见开关，abort 与 session-server 组合被参数校验拒绝。
- e2e oracle 可信度：作者在 Review Focus 中强调 one-token oracle 必须精确复现最终 token轨迹（oracle_output_ids==[final_token_ids[-1]]），否则后续R3字节比对无意义。
  - use_addition_r3 推导与 abort 模式互斥 (design): 采用内部推导 + 启动期校验拒绝 abort，避免不兼容组合静默运行。
  - append 边界设计：checkpoint 长度 vs LCP (design): 采用 checkpoint 长度作为 routed_experts_start_len，LCP 断言前缀稳定，二者职责分离。
  - e2e oracle 可靠性 (testing): 测试显式断言 oracle 输出与最终 token 一致后再进行字节级比对，e2e 通过。

风险与影响

- 风险：
  1. 核心路径变更：chat_completions 主链路新增偏移计算与断言，collect_samples 分支选择 merge 路径；非 addition 模式（默认）行为不变，但 in_place 模式属于每日训练主路径。
  2. 依赖下游 SGLang 分支：routed_experts_start_len 与 --enable-return-routed-experts 依赖修改版 SGLang，若下游不支持该字段可能被忽略或报错，e2e 已用真实 SGLang 验证但生产分支差异仍需注意。
  3. 断言失败路径不对称：_maybe_request_addition_r3 的 assert 在 session 锁内执行，chat_completions 没有像 collect_samples 那样的 422 兜底，一旦前缀稳定性假设被打破会以 500 暴露；测试覆盖了正常与 rollback 场景，但未覆盖断言触发路径。
  4. 参数校验 breaking：--use-session-server --pause-generation-mode=abort 此前可运行，现在启动即失败，存量脚本需适配。
  5. 组装校验严格性：merge_samples_with_addition_r3 对空洞、缺失 patch、行数不匹配全部抛 ValueError 走 422，拒绝产出损坏张量，但任何漏配会造成训练数据断供。 - 影响：启用方式为 --use-session-server --use-rollout-routing-replay --pause-generation-mode in_place，默认关闭；retract 与未开启 session-server 的路径完全不受影响。对 in_place 长会话用户，每轮 R3 传输量从整段前缀降为新增行，显著降低带宽与 SGLang 计算压力，同时 samples 输出与 full-R3 逐字节一致，训练侧无感知。影响模块集中在 session server v1/v2 路由、samples 组装与参数校验，团队需要知晓 abort 组合的校验变化及对下游 SGLang 分支的版本依赖。 - 风险标记：核心路径变更，依赖下游 SGLang 分支，参数互斥新校验，断言失败路径无 422 兜底，组装校验严格性

关联脉络

- PR #2368 fix(rollout): group session v2 leaf samples: 同一 session v2 样本组装链路：多叶子样本共享 rollout_id 与本 PR 的逐叶 R3 物化同属 SessionCoreV2 装配演进，两者互为上下文。