PR #1926 完整报告

radixark/miles

Fix weight sync selector for frozen speculative drafts

合并时间：2026-07-30 08:34

原文链接：http://prhub.com.cn/radixark/miles/pull/1926

执行摘要

本 PR 修复了 raw 转换模式下冻结的 speculative draft 被错误纳入权重更新会话的静默 bug。新增 weight_update_selector 作为唯一决策点，将 "target" 选择器透传到 begin_weight_update 与所有 weight payload，避免对已 shuffle 的 draft 权重二次应用非幂等 AITER shuffle。修复后 spec_accept_length 从 1.00 恢复到 2.75，rollout 时间降低约 34%。改动集中于 miles/backends 下权重同步路径，共 5 个文件，+40/-9。

功能与动机

在 raw conversion 模式下，trainer 可以启用 speculative decoding 而不训练 MTP 层。此时 speculative draft 冻结，不应接收权重更新。但 Miles 之前未告知 SGLang 权重更新覆盖哪些 runner，更新会话和 weight payload 都使用默认 selector "all"，导致每个训练步骤 SGLang 都会对 frozen draft 执行 begin_weight_update / end_weight_update 的 restore 与 finalize 逻辑。因为 draft 的 loader 会过滤所有传入 tensor，所以不会加载新权重，但 finalize 阶段会对已 shuffle 的旧权重再次应用 AITER shuffle_weight。该 shuffle 非幂等，二次应用产生错误排列，draft 推理 kernel 读取错误布局，生成错误预测，且不报错、不崩训练。

PR body 给出了最小复现（shuffle_weight(shuffle_weight(w)) != shuffle_weight(w)）以及端到端对比：frozen-draft reference spec_accept_length = 2.7056，修复前权重同步后跌到 1.00，修复后回到 2.7490。

实现拆解

1. 新增 selector 决策函数（miles/backends/megatron_utils/update_weight/common.py） - 新增 weight_update_selector(args)，返回 "target" 需同时满足：
   - sglang_speculative_algorithm 非空（启用了 speculative decoding）；
   - mtp_num_layers 缺失或为 0（没有 MTP 层需要同步）；
   - megatron_to_hf_mode 不等于 "bridge"（bridge 模式走不同路径）。
   - 该函数成为判断“draft 是否冻结”的唯一位置，避免各调用点各自实现导致不一致。
2. 扩展更新会话入口（common.py + sglang_engine.py） - begin_weight_update(rollout_engines, selector="all") 将 selector 传给每个引擎的 Ray remote 调用。 - SGLang HTTP 客户端的 begin_weight_update 方法增加 selector 参数并写入 payload。

3. 透传 selector 到所有 weight payload（sglang_engine.py） - update_weights_from_tensor 与 update_weights_from_distributed 均增加 selector: str = "all" 参数，并放入请求体，确保实际加载范围与已开启的会话一致。
4. 接入两条更新路径 - colocate/tensor 路径（update_weight_from_tensor.py）：update_weights 中调用 begin_weight_update 时传入 weight_update_selector(self.args)；_send_base_params、_send_lora_params、_send_to_colocated_engine 均透传 selector。 - distributed 路径（update_weight_from_distributed/mixin.py 和 broadcast.py）：_pause_and_prepare_engines 缓存 self._weight_update_selector，在 _update_weight_implementation 调用 update_weights_from_distributed 时传入；broadcast.py 的函数定义同步增加 selector 参数并透传到每个引擎的 remote 方法。
5. 测试与验证 - 未新增自动化测试；依赖 PR body 中的 4 节点端到端对比验证正确性。 - 第二个 commit 仅做 isort 格式整理，无逻辑变化。

miles/backends/megatron_utils/update_weight/common.py

新增 weight_update_selector 并改造 begin_weight_update，是本 PR 的核心决策点与入口。

```
def weight_update_selector(args) -> str:
    """决定本次权重同步覆盖哪些 SGLang runner。

    只有能证明 draft 冻结时才排除 draft，返回 "target"；
    否则保守返回 "all"（target + draft 都覆盖）。
    """
    if (
        getattr(args, "sglang_speculative_algorithm", None) # 启用了 speculative decoding
        and not getattr(args, "mtp_num_layers", None) # 没有训练 MTP 层
        and getattr(args, "megatron_to_hf_mode", "raw") != "bridge" # 非 bridge 转换路径
    ):
        return "target"
    return "all"


def begin_weight_update(rollout_engines: Sequence[ActorHandle], selector: str = "all"):
    """在选中的 rollout 引擎上开启权重更新会话。

    selector 必须与后续 weight payload 保持一致，
    避免更新会话覆盖范围与实际加载范围不一致。
    """
    ray.get([engine.begin_weight_update.remote(selector=selector) for engine in rollout_engines])
```

评论区精华

Review 无实质讨论，两位 reviewer 直接批准（yueming-yuan APPROVED，guapisolo LGTM）。值得提炼的是 PR body 中的技术论证：

> 每次权重更新也处理了冻结的 draft，即使 draft 没有接收新权重。target 接收未 shuffle 的新权重，因此应用 AITER shuffle 是正确的；draft 保留已 shuffle 的旧权重，SGLang 对其第二次应用 shuffle。由于 shuffle 非幂等，这会产生不同的权重排列，推理 kernel 以

> 错误布局读取权重。
>
> 对照数据：frozen-draft reference 2.7056，weight sync without fix 1.00，with fix 2.7490。

风险与影响

- 依赖 SGLang 版本：selector 字段需要 SGLang 侧支持，否则可能被忽略或报错。_check_weight_sync_results 能捕获显式失败，但静默忽略仍可能保留旧行为。
- 条件判断脆弱：weight_update_selector 依赖 mtp_num_layers 和 megatron_to_hf_mode 参数，配置边界变化可能误判。
- 缺少单元测试：核心逻辑无自动化测试，目前仅靠手工端到端验证。
- 影响两条路径：colocate 与 distributed 同时修改，LoRA 路径也透传了 selector，任何遗漏都可能导致 session 与 payload 范围不一致。

影响用户为 raw 转换 + speculative decoding + 不训 MTP 的配置；修复后恢复预期的 draft 接受率，相对无 MTP 场景 rollout 时间降低约 34%。

关联脉络

- PR #2028（session: collect speculative-decoding counters）与本 PR 同属 speculative-decoding 优化线，前者补齐 rolloutsession 的 spec 计数器，为本 PR 的 spec_accept_length 验证提供了指标基础。
- PR #1928（DSA indexer on Blackwell weight sync）同为权重同步正确性问题，关注 SGLang runner 侧权重格式处理，与本 PR 共享“权重同步覆盖 / 处理范围”这一维护主题。
- 整体演进方向：Miles 在持续收敛训练→rollout 权重同步的边界（LoRA 跳过 base、frozen draft 排除、indexer 量化策略），本 PR 是其中针对 speculative decoding 冻结 draft 的关键修复。