

PR #36934 完整报告

sgl-project/sglang

[Fix] Drop the duplicated DSpark draft sample_block call

合并时间: 2026-08-29 09:10

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/36934>

执行摘要

- 一句话: 修复 DSpark 重复采样, 删除多余 sample_block 调用
- 推荐动作: 该 PR 小而精, 值得快速合入。它修复了一个隐蔽的逻辑错误, 且通过相关测试验证。关注点在于测试覆盖, 建议后续补充一个直接针对该路径的单元测试, 以防回归。

功能与动机

PR 描述指出 `DsparkDraftSampler.__call__` 中 `sample_block` 被调用了两次: 一次在 `if draft_tokens is None:` 块内 (#33561 引入), 另一次在其后, 这会覆盖前一次的结果。导致 `self.out` 和 `self.corrected_out` 分别来自两次独立的采样, 产生不一致的 draft tokens 和 corrected logits。此外还激活了 fused greedy 快速路径, 但其结果被计算后又被丢弃。

实现拆解

1. 移除重复调用: 删除函数尾部冗余的 `sample_block` 调用 (原先在 `else` 分支中), 确保每次调用只执行一次。
2. 统一参数传递: 将 `if draft_tokens is None:` 块内的 `sample_block` 调用的 `hidden_states` 参数从 `hidden_states.view(bs, self.gamma, -1)` 改为 `sample_hidden`, 与之前被删除的调用保持一致, 保证行为正确。
3. 调整控制流: 由于删除了 `else` 分支, `sampler` 变量只在 `if draft_tokens is None:` 块内定义, 因此移除了 `else: sampler = greedy_step_sampler` 分支, 简化逻辑。
4. 测试配套: 未直接修改测试文件, 但通过 `/rerun-test` 触发了相关测试, 包括 `test_basic_sanity_dspark.py`、`test_dspark_kernel_parity.py` 等。

关键文件:

- `python/sglang/srt/speculative/dspark_components/dspark_draft_sampler.py` (模块 DSpark; 类别 source; 类型 core-logic): 核心修复文件, 移除了重复的 `sample_block` 调用, 并修正参数传递。

关键符号: `DsparkDraftSampler.call`

关键源码片段

`python/sglang/srt/speculative/dspark_components/dspark_draft_sampler.py`

核心修复文件, 移除了重复的 `sample_block` 调用, 并修正参数传递。

dspark_draft_sampler.py 中 DsparkDraftSampler.__call__ 的关键部分（修复后）
之前这里有两处 sample_block 调用，导致采样重复；修复后只保留一次。

```
if draft_tokens is None:
    # 根据 folded_sampling 选择采样器
    if self.folded_sampling:
        # 使用带噪声的采样器，支持图内重放
        def sampler(step_logits, step_idx):
            del step_idx
            noise = self.exp_noise[:bs].exponential_()
            return SampleStepTokens.execute(
                step_logits=step_logits,
                temperatures=self.temperatures[:bs],
                greedy_mask=self.greedy_mask[:bs],
                exp_noise=noise,
            )
        else:
            sampler = greedy_step_sampler

    # 只执行一次采样，同时返回 draft tokens 和 corrected logits
    draft_tokens, corrected_logits = self.markov_head.sample_block(
        base_logits,
        first_prev_tokens=anchor,
        hidden_states=sample_hidden,
        sampler=sampler,
        collect_corrected=self.folded_sampling,
    )
    if self.folded_sampling:
        # 只有在 folded sampling 时才需要 corrected logits
        self.corrected_out[: bs * self.gamma].copy_(
            corrected_logits.reshape(bs * self.gamma, -1)
        )

# 写入输出缓冲区
self.out[: draft_tokens.numel()].copy_(draft_tokens.reshape(-1))
```

评论区精华

无正式的 review 评论，但 PR 作者在 Issue 评论中触发了相关测试的 rerun，实验结果均为通过。

- 暂无高价值评论线程

风险与影响

- 风险：回归风险：修改了核心采样逻辑，可能导致 DSpark 相关测试失败或行为变化，但由于测试已通过，风险较低。一致性风险：修复后，`corrected_logits` 和 `draft_tokens` 将由同一次采样产生，消除了之前的不一致。性能：消除了重复计算，提升了性能，但影响范围有限。

- 影响：用户：修复了 DSpark 采样可能产生不一致 draft 和 corrected logits 的 bug，影响服务正确性，但仅影响 DSpark 相关场景。系统：减少了重复计算，轻微提升性能，影响限于 DSpark 模块。团队：代码更简洁，逻辑更清晰，便于后续维护。
- 风险标记：核心路径变更，缺少直接单元测试

关联脉络

- PR #33561 Draft sampler sample_block call: 该 PR 在 if draft_tokens is None 块内引入了 sample_block 调用，本 PR 修复了由此引入的重复调用问题。