

PR #38585 完整报告

sgl-project/sglang

[AMD][CI] Skip failing Wave test and relax multi-LoRA output check

合并时间: 2026-09-12 03:25

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/38585>

执行摘要

- 一句话: 跳过 AMD Wave 测试并放宽多 LoRA 容差
- 推荐动作: 值得快速浏览 (约 10 分钟)。本 PR 虽只是测试配套, 但 review 中 yctseng0211 用纯 transformers 扫描 batch size 定位参考文本漂移的方法, 是排查“实现回归 vs 环境漂移”的经典范例, 对维护多硬件 CI 的团队有借鉴价值。值得关注的设计决策: 把放宽范围用 `is_in_amd_ci()` 严格限定到特定环境而不是全局放宽; 以及用 `@unittest.skip` 保留用例体、便于内核修复后一键恢复的做法。建议为 Wave NaN 根因补一个跟踪 issue, 避免跳过变成永久跳过。

功能与动机

ROCm 10 下两个测试持续阻塞 AMD CI: 一是 Wave grouped-decode 测试返回 NaN; 二是 Llama-2 multi-LoRA 测试的无 adapter 对照组中, Hugging Face 参考文本在 ROCm 10 上出现轻微漂移 (ROUGE-L 0.9774, 差一个词), 而两个 LoRA adapter 的输出本身仍然匹配。PR body 明确指出“The strict 1.0 check therefore blocks AMD CI without showing a LoRA regression”——严格检查并未暴露 LoRA 回归, 反而挡住整条 CI。

实现拆解

变更入口是两条测试通道: AMD Wave attention 内核单测与 LoRA 端到端对比测试。按以下步骤拆解:

1. 跳过 Wave grouped-decode 测试: 在 `test/registered/attention/test_wave_attention_kernels.py` 中为 `test_grouped_decode_attention` 添加 `@unittest.skip`, 注明“Wave grouped decode is temporarily disabled because it produces NaNs on ROCm 10”。原有的 4 种序列长度 × 6 种 (B, H_Q, H_KV, D, D_V) 配置循环体原地保留, 便于内核修复后直接摘除装饰器恢复覆盖。
2. 引入 AMD CI 环境探测: `python/sglang/test/lora_utils.py` 的导入从 `sglang.test.test_utils` 扩展出 `is_in_amd_ci()`, 作为容差门控的环境开关。
3. 多 LoRA 容差环境门控: 在 `CI_MULTI_LORA_MODELS` 的 `meta-llama/Llama-2-7b-hf` 用例 (no-adapter 对照组) 上新增 `rouge_l_tolerance=0.9 if is_in_amd_ci() else 1.0`, 把放宽严格限定在 AMD CI; 两个 adapter 原本就有的 `rouge_l_tolerance=0.9` 不受影响。CUDA 及其他环境仍要求 ROUGE-L 1.0 的精确匹配。

4. Review 迭代收敛：4 个 commit 展示了从初始放宽到收敛的过程——“Scope multi-LoRA tolerance to AMD CI”把放宽范围限定到 AMD，“Simplify multi-LoRA tolerance gate”简化门控写法，最终由 HaiShaw approve 合并。

无生产源码、配置或部署配套改动；全部改动落在测试代码，属于 CI 稳定化配套。

关键文件：

- python/sglang/test/lora_utils.py（模块 LoRA 工具；类别 test；类型 test-coverage）：多 LoRA 对比测试的容差门控所在文件：新增 is_in_amd_ci() 导入，并在 CI_MULTI_LORA_MODELS 的 Llama-2 no-adapter 对照组上新增环境条件容差，是本 PR 的核心逻辑改动。
- test/registered/attention/test_wave_attention_kernels.py（模块 注意力内核；类别 test；类型 test-coverage；符号 test_grouped_decode_attention）：Wave attention 内核单测：test_grouped_decode_attention 因 ROCm 10 产生 NaN 被挂上 @unittest.skip，跳过决策直接影响 AMD 上该内核路径的测试覆盖。

关键符号：test_grouped_decode_attention, is_in_amd_ci

关键源码片段

python/sglang/test/lora_utils.py

多 LoRA 对比测试的容差门控所在文件：新增 is_in_amd_ci() 导入，并在 CI_MULTI_LORA_MODELS 的 Llama-2 no-adapter 对照组上新增环境条件容差，是本 PR 的核心逻辑改动。

```
# python/sglang/test/lora_utils.py 中的多 LoRA CI 用例配置
# 关注点：no-adapter 对照组的 ROUGE-L 容差被做成了环境门控。
# 仅当 `is_in_amd_ci()` 为真（AMD CI 流水线）时放宽到 0.9,
# 其余环境（CUDA 等）保持 1.0 精确匹配，避免全局降低测试灵敏度。
CI_MULTI_LORA_MODELS = [
    # multi-rank case: 两个 adapter 各自仍保留原有的 0.9 ROUGE-L 容差
    LoRAModelCase(
        base="meta-llama/Llama-2-7b-hf",
        adaptors=[
            LoRAAdaptor(
                name="winddude/wizardLM-LLaMA-LoRA-7B",
                prefill_tolerance=1e-1,
                rouge_l_tolerance=0.9,
            ),
            LoRAAdaptor(
                name="RuterNorway/Llama-2-7b-chat-norwegian-LoRa",
                prefill_tolerance=3e-1,
                rouge_l_tolerance=0.9,
            ),
        ],
    ),
    # 无 adapter 对照组的容差：
    # ROCm 10 / torch 2.11 下 HF 参考文本在 bs=1 时选词漂移（ROUGE-L 约 0.978），
    # 而 HFRRunner 单条跑、SRTRunner 批量跑 3 条，batch 维度差异放大了漂移；
```

```

        # 这里只是放宽无 adapter 基线, adapter 的输出仍被 0.9 容差校验。
        rouge_l_tolerance=0.9 if is_in_amd_ci() else 1.0,
        max_loras_per_batch=2,
        max_loaded_loras=4,
    ),
]

```

test/registered/attention/test_wave_attention_kernels.py

Wave attention 内核单测: `test_grouped_decode_attention` 因 ROCm 10 产生 NaN 被挂上 `@unittest.skip`, 跳过决策直接影响 AMD 上该内核路径的测试覆盖。

```

# test/registered/attention/test_wave_attention_kernels.py
# ROCm 10 上 Wave grouped decode 内核会输出 `NaN`, 临时禁用该用例。
# 注意: 这是跳过而非修复, 内核 `NaN` 根因仍未解决,
# 跳过期间 AMD 上该内核路径失去 CI 守护, 需跟踪后续内核修复以恢复覆盖。
@unittest.skip(
    "Wave grouped decode is temporarily disabled because it produces NaNs on ROCm 10."
)
def test_grouped_decode_attention(self):
    # 覆盖 4 种序列长度与 6 种形状组合 (B, H_Q, H_KV, D, D_V),
    # 每种组合都对比 Wave 内核与 Triton 参考实现的输出 (cosine > 0.99 且 allclose)。
    seq_lens = [5, 100, 128, 500]
    configs = [
        (2, 16, 16, 64, 64),
        (2, 16, 1, 64, 64),
        (2, 128, 1, 80, 80),
        (32, 128, 2, 512, 512),
        (2, 128, 2, 512, 512),
        (2, 128, 1, 576, 512),
    ]
    for S in seq_lens:
        for B, H_Q, H_KV, D, D_V in configs:
            self._test_grouped_decode_attention_once(B, S, H_Q, H_KV, D, D_V)

```

评论区精华

核心争论围绕“要不要放宽、放宽多少”展开, 最终以环境限定 + 监控的方式收敛:

- HaiShaw 在 `python/sglang/test/lora_utils.py:154` 直接追问 “why is this?”, 质疑无 adapter 对照组的容差从 1.0 降到 0.9 是否合理。
- bingxche 答复: ROCm 10 上该用例得分为 0.978, 与 HF 结果只差一个词 (“massive amounts / a massive amount”), 需要稍微放宽以解除 CI 阻塞, 并持续监控是否出现严重回归。
- yctseng0211 给出关键溯源: 用纯 transformers (无 SGLang、无 LoRA) 扫描 batch size 后发现, 两个软件栈都会产生两种措辞变体; 决定测试结果的是 bs=1 这一格, 因为 `HFRunner` 一次只跑 1 个 prompt, 而 `SRTRunner` 批量跑 3 个。ROCm 10 / torch 2.11 下 bs=1 与 8 选 “massive amounts”, ROCm 7.2 / torch 2.9.1 下 bs=1 选 “a massive amount”——漂移来自参考文本对 batch 维度的敏感性, 与 LoRA 实现无关。

- 为何放宽 no-adapter 对照组的 ROUGE-L 容差 (question): 以环境限定方式 (仅 AMD CI 放宽到 0.9, 其余环境保持 1.0) 达成共识, 合并时由 HaiShaw approve。
- 参考文本漂移的根源: batch 维度而非 LoRA 回归 (correctness): 该分析确认 0.9774 漂移属于参考文本对 batch 的敏感性问题, 不是 LoRA 回归, 支持了放宽决策的合理性。

风险与影响

- 风险:
 1. 测试跳过掩盖真实内核缺陷: Wave grouped-decode 在 ROCm 10 上产生 NaN 属于内核正确性问题, 本次仅加 @unittest.skip 而未修复根因, AMD 上该路径将暂时失去 CI 守护; 若 ROCm 10 成为 AMD 默认运行环境, NaN 风险可能漏到运行时, 建议用 issue 跟踪根因修复并恢复测试。
 2. 容差放宽降低回归灵敏度: no-adapter 对照组降到 0.9 后, ROUGE-L 在 0.9 到 1.0 区间内的轻微 LoRA 回归可能被放过。但两个 adapter 的自身容差原本就是 0.9, 本次只是统一了对照组口径, 且实际分数 0.978 距阈值 0.9 仍有 0.078 余量, 足以捕获明显回归。
 3. 环境判断依赖 CI 变量: is_in_amd_ci() 若在非 AMD 环境误判为真, 会破坏 CUDA 等环境的严格匹配; 代码默认分支为 1.0, 误判面较小。
 4. 影响面有限: 无生产代码改动, 风险集中在测试有效性而非运行时行为。
- 影响:
 - 对 CI: AMD ROCm 10 流水线由红转绿, 解除 Wave 测试和多 LoRA 测试的双重阻塞; CUDA 与其他环境的严格 1.0 检查保持不变。
 - 对测试资产: Wave grouped-decode 用例被跳过但保留在代码中, 便于恢复; 多 LoRA 测试的对照组容差成为环境敏感配置, 后续维护需留意 is_in_amd_ci() 的语义。
 - 对团队: 形成“环境限定容差 + 跳过并跟踪”的 CI 稳定化模式, 后续 ROCm/AMD 相关测试可复用同一门控函数。
 - 对用户: 无任何运行时影响。
 - 风险标记: 测试跳过掩盖内核缺陷, 容差放宽降低回归灵敏度, 环境判断依赖 CI 变量, 缺少根因跟踪

关联脉络

- PR #39044 [AMD][CI] Temporarily pause MI355X disaggregated nightly: 同为 AMD CI 临时稳定化手段 (暂停 vs 跳过 / 放宽), 反映 AMD CI 在 ROCm 环境切换期的系统性不稳定。
- PR #38756 [AMD] aiter: resolve SWA KV pool for draft workers + guard paged decode: 同为 AMD attention 后端 / 内核的正确性守卫与测试配套, 属于同一硬件关注线。
- PR #38936 [Fix] Disable NCCL graph buffer registration for the TP LM-head all-to-all (pure-DP decode hang under request bursts): 同样以“禁用 / 降级某个能力”换取运行时或 CI 稳定性的处理模式, 可对比两者对能力损失的跟踪方式。