PR #23517 完整报告

sgl-project/sglang

[diffusion] CI: do not retry consistency failures

合并时间：2026-04-23 12:39

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/23517

执行摘要

- 一句话：修改扩散模型测试套件，一致性检查失败时不再重试。
- 推荐动作：该 PR 变更简单直接，适合快速浏览以了解测试重试策略的调整。值得关注的设计决策是：将一致性检查失败明确排除在重试逻辑之外，这反映了对失败类型的分类处理（瞬态 vs. 非瞬态），有助于优化 CI 资源使用。

功能与动机

PR 标题和提交信息直接表明动机是“CI: do not retry consistency failures”。虽然没有详细的 PR body 描述，但从变更内容可以推断，目的是防止 CI 在遇到扩散模型测试中的一致性检查失败（例如“Consistency check failed for”或“GT not found for”）时进行重试，因为这些失败通常不是由瞬态问题（如网络超时、内存不足）引起的，重试无法解决问题，反而浪费CI资源。

实现拆解

1. 新增一致性检查失败检测函数：在 python/sglang/multimodal_gen/test/run_suite.py 中新增 _is_consistency_failure 函数，通过解析测试输出摘要和完整输出来识别包含特定错误模式（如“Consistency check failed for”、“GT not found for”、“--- MISSING GROUND TRUTH DETECTED ---”）的失败。
2. 修改可重试失败判断逻辑：在同一个文件的 _is_retryable_failure 函数开头添加条件检查，如果 _is_consistency_failure 返回 True，则直接返回 False，表示该失败不可重试。这确保了 CI 不会对一致性检查失败进行重试。
3. 测试配套调整：此变更属于测试基础设施的调整，没有修改核心模型或运行时逻辑，仅影响测试运行时的重试行为。

关键文件：

- python/sglang/multimodal_gen/test/run_suite.py（模块 扩散测试；类别 test；类型 test-coverage；符号 _is_consistency_failure, _is_retryable_failure）：这是唯一被修改的文件，包含了测试套件中重试逻辑的核心调整，直接影响 CI 对扩散模型测试失败的处理方式。

关键符号：_is_consistency_failure, _is_retryable_failure

关键源码片段

python/sglang/multimodal_gen/test/run_suite.py

这是唯一被修改的文件，包含了测试套件中重试逻辑的核心调整，直接影响 CI 对扩散模型测试失败的处理方式。

```
def _is_consistency_failure(full_output: str) -> bool:
    """
    检测测试输出中是否包含一致性检查失败的错误信息。
    这类失败通常是由于缺少基准真值（Ground Truth）或数据不一致引起的，
    不属于瞬态问题（如OOM、超时），因此不应在CI中重试。
    """
    summary_lines = _extract_short_test_summary(full_output)
    for line in summary_lines:
        if "Consistency check failed for" in line or "GT not found for" in line:
            return True # 在摘要行中找到一致性失败模式

    return (
        "Consistency check failed for " in full_output
        or "GT not found for " in full_output
        or "--- MISSING GROUND TRUTH DETECTED ---" in full_output
    ) # 在完整输出中检查其他相关模式


def _is_retryable_failure(full_output: str) -> bool:
    """
    判断测试失败是否可重试。如果是一致性检查失败，则直接返回False，
    避免CI对非瞬态问题进行无效重试。
    """
    if _is_consistency_failure(full_output):
        return False # 一致性失败不可重试

    # 原有的重试逻辑保持不变，处理其他类型的失败（如性能断言、OOM 等）
    summary_lines = _extract_short_test_summary(full_output)
    is_perf_assertion = (
        "multimodal_gen/test/server/test_server_utils.py" in full_output
        and "AssertionError" in full_output
    )
    is_aggregated_retryable_failure = _summary_has_retryable_failure(summary_lines)
    is_flaky_ci_assertion = (
        "SafetensorError" in full_output
        or "FileNotFoundError" in full_output
        or "TimeoutError" in full_output
    )
    is_oom_error = (
        "out of memory" in full_output.lower() or "oom killer" in full_output.lower()
    )

    return (
        is_perf_assertion
```

```
        or is_aggregated_retryable_failure
        or is_flaky_ci_assertion
        or is_oom_error
    )
```

评论区精华

该PR没有review评论或讨论，表明变更直接、无争议，可能由熟悉该模块的维护者快速合并。

- 暂无高价值评论线程

风险与影响

- 风险：技术风险较低：
  - 回归风险：修改仅限于测试重试逻辑，不影响扩散模型的核心功能或性能。但如果 _is_consistency_failure 函数误判（例如将其他类型的失败错误地标记为一致性失败），可能导致本应重试的瞬态失败被跳过，增加 CI 假阴性风险。不过，函数使用的错误字符串模式（如"Consistency check failed for"）相对特定，误判可能性较小。
  - 兼容性：无兼容性问题，因为不涉及 API 或数据格式变更。
  - 性能：无性能影响，仅增加少量字符串匹配开销，在测试上下文中可忽略。
- 影响：影响范围有限但直接：
  - 对用户：无直接影响，因为这是内部 CI 逻辑调整。
  - 对系统：优化 CI 行为，减少对非瞬态失败的无意义重试，可能缩短 CI 运行时间并节省计算资源。
  - 对团队：提高 CI 效率，避免因一致性检查失败导致的重复测试运行，但需要确保开发人员理解此类失败不会被自动重试，可能需要手动干预。
  - 风险标记：误判风险

关联脉络

- PR #23198 [diffusion] Fix --warmup-resolutions hang with --enable-cfg-parallel: 同属扩散模块，涉及测试或运行时问题修复，可能共享相似的 CI 上下文。
- PR #22953 [diffusion][bugfix] avoid illegal memory access in qwen image: 同属扩散模块，关注测试或模型中的具体问题，体现该模块的持续维护。