

PR #29831 完整报告

sgl-project/sglang

[diffusion] Prefer official diffusion consistency GT

合并时间: 2026-07-04 10:09

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/29831>

执行摘要

- 一句话: 优先使用官方一致性 GT, 缺失时回退到 sglang 版本
- 推荐动作: 建议开发者阅读 test_utils.py 中的 _load_official_consistency_gt_outputs 和 _remote_consistency_gt_base_urls 实现, 这是一个优雅的多源 GT 优先级策略。测试编写者也值得学习如何用 monkeypatch 模拟远程依赖来覆盖复杂分支。

功能与动机

在 diffusion 模型一致性测试中, 官方仓库 (如原始模型仓库) 现在也生成了 GT 输出, 这些输出被视为更权威的基准。优先使用官方 GT 可以更早发现回归问题, 并提高测试的可信度。PR 说明中也提到要保持 Ascend 专用 GT 查找在前, 并使用单元测试覆盖选择顺序。

实现拆解

1. 动态加载官方 GT 映射: 在 test_utils.py 中引入 _load_official_consistency_gt_outputs 函数, 通过 HTTP GET 请求远程 case_map.json 获取每个 case 声明的官方输出文件列表, 结果缓存到全局变量 _official_consistency_gt_outputs_cache。
2. 构建多源 GT 优先级逻辑: 新增 _remote_consistency_gt_base_urls 函数 (替代原硬编码的 SGL_TEST_FILES_CONSISTENCY_GT_BASES), 根据平台 (普通或 Ascend) 和 case 是否声明了官方 GT, 动态决定 base URL 顺序: 先官方 base, 再 sglang base。Ascend 平台的专用 URL 始终优先于默认目录。
3. 处理跳过和未声明 case: 定义 OFFICIAL_CONSISTENCY_GT_SKIP_CASES 集合, 跳过已知有问题的 LTX 和 Qwen edit case, 强制使用 sglang GT。case 若未在 case_map.json 中声明, 则忽略官方目录, 直接走 sglang 路径。
4. 更新 CI 数据修订版: 将 SGL_TEST_FILES_CI_DATA_REVISION 更新到包含官方 GT 的新提交。
5. 单元测试覆盖: 在 test_consistency_metrics.py 中通过 monkeypatch 模拟远程响应, 验证: 优先选择官方文件、忽略未映射文件、回退行为、跳过集合、视频场景等。
6. 调整一致性阈值: 修改 h100.json 中多个 case 的 PSNR、SSIM 和 mean_abs_diff 阈值, 使其适合与官方 GT 比较。
7. 附带改进: 修复 server_warmup.py 的日志格式 (失败时记录堆栈), 调整 gen_diffusion_ci_outputs.py 的导入顺序。

关键文件:

- python/sglang/multimodal_gen/test/test_utils.py (模块 测试工具; 类别 test; 类型 test-coverage; 符号 _load_official_consistency_gt_outputs, _official_consistency_gt_outputs_for_case, _is_official_consistency_gt_base_url, _official_consistency_gt_candidate_is_declared) : 实现核心 GT 选择逻辑, 包括动态加载官方 GT 映射、优先级排序、Ascend 平台处理、跳过集合
- python/sglang/multimodal_gen/test/unit/test_consistency_metrics.py (模块 测试用例; 类别 test; 类型 test-coverage; 符号 _set_official_gt_outputs, _disable_remote_official_gt_case_map, test_remote_image_gt_prefers_official_when_present, test_remote_image_gt_ignores_unmapped_official_file) : 添加大量单元测试覆盖官方 GT 优先、回退、跳过、Ascend 平台等场景
- python/sglang/multimodal_gen/test/server/consistency_thresholds/h100.json (模块 配置; 类别 test; 类型 configuration) : 调整阈值以匹配新官方 GT, 影响测试通过标准
- python/sglang/multimodal_gen/runtime/server_warmup.py (模块 预热器; 类别 source; 类型 core-logic) : 附带日志改进, 失败时记录完整堆栈
- python/sglang/multimodal_gen/test/server/perf_baselines/h100.json (模块 配置; 类别 test; 类型 configuration) : 调整 e2e 性能基线以匹配新 GT 和阈值
- python/sglang/multimodal_gen/test/scripts/gen_diffusion_ci_outputs.py (模块 CI 脚本; 类别 test; 类型 other) : 调整导入顺序, 适应重构

关键符号: _load_official_consistency_gt_outputs, _official_consistency_gt_outputs_for_case, _is_official_consistency_gt_base_url, _official_consistency_gt_candidate_is_declared, _remote_consistency_gt_base_urls, _is_ascend_consistency_case, _find_remote_consistency_gt_files, _set_official_gt_outputs, get_consistency_gt_candidates

关键源码片段

python/sglang/multimodal_gen/test/test_utils.py

实现核心 GT 选择逻辑, 包括动态加载官方 GT 映射、优先级排序、Ascend 平台处理、跳过集合

```
def _load_official_consistency_gt_outputs() -> dict[str, frozenset[str]]:
    """Return case_id -> declared official GT outputs from the pinned ci-data map."""
    global _official_consistency_gt_outputs_cache
    if _official_consistency_gt_outputs_cache is not None:
        return _official_consistency_gt_outputs_cache

    # 构造远程 case_map.json URL, 使用固定修订版避免缓存不一致
    url = f"{SGL_TEST_FILES_OFFICIAL_CONSISTENCY_GT_BASE}/case_map.json"
    outputs_by_case: dict[str, frozenset[str]] = {}
    try:
        resp = requests.get(url, timeout=30)
        if resp.status_code == 200:
            data = resp.json()
            # 转换每个 case 的输出文件列表为 frozenset 以便快速查找
```

```

        for case_id, outputs in data.items():
            outputs_by_case[case_id] = frozenset(outputs)
        else:
            logger.warning("Failed to load official consistency GT case map, HTTP %d", resp.status_code)
    except Exception as exc:
        logger.warning("Failed to load official consistency GT case map: %s", exc)

    _official_consistency_gt_outputs_cache = outputs_by_case
    return outputs_by_case

```

python/sglang/multimodal_gen/test/unit/test_consistency_metrics.py

添加大量单元测试覆盖官方 GT 优先、回退、跳过、Ascend 平台等场景

```

def test_remote_image_gt_prefers_official_when_present(monkeypatch):
    # 设置平台为 h100
    monkeypatch.setenv(test_utils.CONSISTENCY_PLATFORM_ENV, "h100")
    # 构造一个匹配的官方文件
    expected_filename = f"unit_image_1gpu.{test_utils.output_format_to_ext(None)}"
    _set_official_gt_outputs(monkeypatch, {"unit_image": [expected_filename]})
    # 模拟只有官方 URL 可访问
    monkeypatch.setattr(test_utils, "_remote_file_exists",
                        lambda url: url.startswith(test_utils.SGL_TEST_FILES_OFFICIAL_
                                                    CONSISTENCY_GT_BASE + "/"))

    files = test_utils._find_remote_consistency_gt_files("unit_image", 1, is_video=False)

    # 断言只返回官方路径
    assert files == [(expected_filename,
                      f"{test_utils.SGL_TEST_FILES_OFFICIAL_CONSISTENCY_GT_BASE}/{expected_filename}")]

```

评论区精华

无实质性讨论，本 PR 由提交者独立演进，未产生 review 对话。自动化代码审查工具（gemini-code-assist）给出了无反馈的评论。

- 暂无高价值评论线程

风险与影响

- 风险：
 1. 远程配置依赖：_load_official_consistency_gt_outputs 依赖 ci-data 仓库的 case_map.json 可用性和网络连通性。如果该服务不可用，测试将失败或回退到 sglang GT（取决于异常处理）。当前实现在请求失败时返回空映射，然后 fallback 到 sglang GT，具有一定韧性。
 2. Skip 列表维护成本：OFFICIAL_CONSISTENCY_GT_SKIP_CASES 中的 case 需要手动维护，如果问题 case 修复后未及时移除，会导致永远不使用官方 GT。

3. 阈值调整影响: h100.json 的阈值同时被放宽和收紧, 可能导致某些 case 更容易通过或失败。如果阈值与官方 GT 不匹配, 可能掩盖回归。
4. 缓存一致性: _official_consistency_gt_outputs_cache 是进程级缓存, 单次测试运行中生效。但若某 case 的官方 GT 中途被更新, 不会重新加载。考虑到测试运行时间较短, 此风险可控。
 - 影响: 影响所有运行 diffusion 一致性测试的 CI 平台 (H100、B200、5090、NPU 等)。开发者将使用更权威的官方 GT 进行回归检测, 提高测试可靠性。需要团队维护远程 case_map.json 和跳过集合, 但维护成本较低。附带日志改进提升了 warmup 失败的可调试性。
 - 风险标记: 远程配置依赖, skip cases 维护成本, 阈值调整影响

关联脉络

- 暂无明显关联 PR