

PR #29926 完整报告

sgl-project/sglang

Fix Diffusion GT generation pipelines

合并时间: 2026-07-05 16:43

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/29926>

执行摘要

- 一句话: 修复 diffusion CI 生成脚本因上游重构而中断
- 推荐动作: 该 PR 是小型修复, 值得快速合并以恢复 CI。但对于未来, 考虑将 `collect_test_items` 放在共享的工具模块中, 以避免重复。

功能与动机

PR #28762 移除了 `run_suite.py` 中的 `collect_test_items` 导出, 导致 `gen_diffusion_ci_outputs.py` 无法导入该函数, 进而破坏了 `diffusion-ci-gt-gen*.yaml` 工作流。本 PR 旨在将缺失的函数直接内联到脚本中, 恢复受影响的工作流。

实现拆解

1. 内联 `collect_test_items` 函数 (`python/sglang/multimodal_gen/test/scripts/gen_diffusion_ci_outputs.py`): 新函数通过 `subprocess.run` 调用 `pytest --collect-only -q` 收集测试节点 ID, 然后解析 `stdout` 筛选出包含 `::` 的测试项。
2. 调整导入语句: 移除原有从 `run_suite` 模块导入的依赖, 并将 `run_pytest` 导入改为 `run_pytest` 仅保持外部依赖不变。
3. 增加 `subprocess` 导入: 新增 `import subprocess` 以支持函数实现。
4. 修复工作流依赖: 此直接内联避免了跨模块依赖, 使脚本独立运行, 确保 `diffusion-ci-gt-gen*.yaml` 工作流恢复功能。

关键文件:

- `python/sglang/multimodal_gen/test/scripts/gen_diffusion_ci_outputs.py` (模块 CI 脚本; 类别 `test`; 类型 `test-coverage`; 符号 `collect_test_items`): 唯一修改的文件, 内联了 `collect_test_items` 函数, 修复了因上游重构导致的导入错误。

关键符号: `collect_test_items`

关键源码片段

`python/sglang/multimodal_gen/test/scripts/gen_diffusion_ci_outputs.py`

唯一修改的文件, 内联了 `collect_test_items` 函数, 修复了因上游重构导致的导入错误。

```
def collect_test_items(files: list[str], filter_expr: str | None = None) -> list[str]:  
    """Collect test node IDs from the given files using pytest --collect-only."""
```

```

cmd = [sys.executable, "-m", "pytest", "--collect-only", "-q"]
if filter_expr:
    cmd.extend(["-k", filter_expr])
cmd.extend(files)

filter_note = f" with filter: {filter_expr}" if filter_expr else ""
print(f"Collecting tests from {len(files)} file(s){filter_note}")
result = subprocess.run(cmd, capture_output=True, text=True)

# exit code 5 means no tests collected (acceptable with filters)
if result.returncode not in (0, 5):
    error_msg = (
        f"pytest --collect-only failed with exit code {result.returncode}\n"
        f"Command: {' '.join(cmd)}\n"
    )
    if result.stderr:
        error_msg += f"stderr:\n{result.stderr}\n"
    if result.stdout:
        error_msg += f"stdout:\n{result.stdout}\n"
    logger.error(error_msg)
    raise RuntimeError(error_msg)

if result.returncode == 5:
    print(
        "No tests were collected (exit code 5). This may be expected with filters."
    )

# Parse pytest output to extract test node IDs (lines containing "::")
test_items = []
for line in result.stdout.strip().split("\n"):
    line = line.strip()
    if line and "::" in line and not line.startswith(("=", "-", " ")):
        test_id = line.split()[0] if " " in line else line
        if "::" in test_id:
            test_items.append(test_id)

print(f"Collected {len(test_items)} test items")
return test_items

```

评论区精华

gemini-code-assist[bot] 建议将 `::` 检查改为 `.py::`，以避免警告中的误匹配。作者 e-martirosian 回复 "Don't need such changes"，认为现有检查足够，最终该建议未采纳。

- pytest 输出解析鲁棒性：建议用 `.py::` 代替 `::` (correctness)：作者 e-martirosian 认为无需改动，现有检查已足够。

风险与影响

- 风险：风险较低。该变更仅涉及内联一个已经存在且测试过的函数，逻辑未改变。但新的实现通过子进程调用 `pytest`，若 `pytest` 输出格式未来变化可能导致解析失败。
- 影响：直接影响 diffusion CI 的 GT 生成 workflow，使其恢复到正常状态。对用户无影响，因为这是 CI 基础设施的一部分。间接影响是降低了 `gen_diffusion_ci_outputs.py` 与 `run_suite.py` 的耦合，使脚本更独立。
- 风险标记：CI 恢复修复

关联脉络

- PR #28762 Remove `collect_test_items` from `run_suite.py`: 本 PR 直接修复了 PR #28762 引入的破坏，该 PR 移除了 `collect_test_items` 的导出。