

PR #24487 完整报告

sgl-project/sglang

propagate pytest exit code from test __main__ entries

合并时间: 2026-05-07 09:46

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/24487>

执行摘要

本 PR 修复了 CI 中因测试文件使用裸露 `pytest.main()` 导致测试失败被静默吞没的问题。通过新增基于 AST 扫描的静态守卫和删除 9 个受影响文件，确保未来不会再有此类回归被掩盖。

功能与动机

CI 运行器 `python/sglang/test/ci/ci_utils.py` 通过包装脚本的 `returncode` 判断测试是否通过。部分测试文件使用 `if __name__ == "__main__": pytest.main([__file__])`，但未用 `sys.exit()` 包装，导致 `pytest` 返回码丢失，脚本始终返回 0。即使测试失败，CI 日志也显示 PASSED。该问题在 `test_norm_tanh_mul_add_norm_scale.py` 的输出 `1 failed, 47 passed` 但 CI 全部变绿时被发现。

实现拆解

- 新增 AST 守卫测试: `test/registered/unit/test_no_bare_pytest_main.py` 使用 Python `ast` 模块扫描 `python/` 和 `test/` 目录下所有 `.py` 文件，检测 `if __name__ == "__main__":` 块内是否包含裸露的 `pytest.main(...)` 调用（即作为表达式语句，未用 `sys.exit()` 包装）。若发现违规，断言失败。关键符号: `_find_bare_pytest_main`、`_is_main_guard`、`_is_bare_pytest_main_call`、`test_no_bare_pytest_main_in_repo`。
- 删除裸露调用文件: 删除 9 个使用裸露 `pytest.main()` 的测试文件，涉及 JIT kernel、multimodal 和 LoRA 模块。这些文件要么已静默失败，要么需要原作者后续用 `sys.exit(pytest.main(...))` 重新添加。在删除前，对 `test_lora_moe_runner.py` 和 `test_norm_tanh_mul_add_norm_scale.py` 进行了补丁修复。
- 保持运行器不变: CI 仍使用 `python3 file.py -f` 方式调用，全局切换至 `python -m pytest` 会影响 696 个 `unittest.main()` 文件，超出范围。

由于材料未提供具体代码片段，此处无法展示。新增守卫的主要逻辑是 AST 遍历并匹配函数调用，删除的文件原入口均为 `if __name__ == "__main__": pytest.main([__file__])`。

评论区精华

Reviewer `DarkSharpness` 直接批准，无实质讨论。实现过程中曾有一个包含冗余行为检查的版本（commit `2bbf631`），后简化为只保留 AST 扫描。

风险与影响

- 测试覆盖损失：删除 9 个测试文件后，JIT kernel 和 LoRA 相关测试覆盖完全移除，原作者重新添加前回归风险上升。
- 守卫假阳性：AST 扫描可能误判合法但非裸露的调用，但设计保守，风险较低。
- CI 时间增加：守卫需遍历仓库，但影响有限。
- 影响范围：仅 CI 内部变更，用户无感知；团队需跟进重新添加删除的测试。

关联脉络

本 PR 与历史 PR 无直接关联，但后续可观察原作者是否提交重新添加已删测试的 PR。