

PR #2561 完整报告

radixark/miles

test(ci): file the verifiers tests under the example they test

合并时间: 2026-08-18 04:10

原文链接: <http://prhub.com.cn/radixark/miles/pull/2561>

执行摘要

- 一句话: verifiers 三个测试按 example 目录归档, 纯移动无内容变更
- 推荐动作: 不值得精读, 属于低风险、高确定性的测试组织维护。值得注意的是其确立的目录镜像原则——“tests/fast/ 下的目录应镜像被测代码的包结构”, 以及 PR body 中展示的“纯移动安全性论证”模板 (包路径访问、无 `__file__`、无外部引用、CI list-only 对比), 可作为后续同类测试迁移 PR 的参照。

功能与动机

PR body 明确指出: `tests/fast/rollout/` 镜像 `miles/rollout/`, 其中其他测试 (`test_fully_async_rollout.py`、`test_checkpoint_eval.py`、`generate_hub/`、`rm_hub/`、`session/`) 都覆盖 miles 自有 rollout 代码, 而 verifiers 的两个测试是仅有的覆盖 example 的文件; 同时 launcher 测试松散放在 `experimental/` 根目录, 与仓库惯例 (`swe_agent_harbor_docker/`、`p2p_weight_transfer/` 均有独立目录) 不符。作者还特别论证了安全性: 三个测试全部通过包路径 `examples.experimental.verifiers...` 访问被测对象, 无需 `conftest` 或 `sys.path` 引导, 且无文件引用 `__file__`, 仓库内也无其他文件引用旧路径。

实现拆解

1. 确定目标目录: 新建 `tests/fast/examples/experimental/verifiers/` 目录, 并新增空 `__init__.py`, 使该目录成为可导入的测试包, 与 `swe_agent_harbor_docker`、`p2p_weight_transfer` 等既有 example 测试目录保持一致。
2. 迁移两个 rollout 相关测试: 将 `tests/fast/rollout/test_verifiers_rollout.py` 与 `tests/fast/rollout/test_verifiers_runtime.py` 原样移动到新目录, 文件名不变。这两个文件均导入 `examples.experimental.verifiers.verifiers_rollout`, 属于 example 测试而非 miles 自有 rollout 代码测试, 移除后 `tests/fast/rollout/` 恢复为纯镜像 `miles/rollout/` 的职责。
3. 重命名并迁移 launcher 测试: 将 `tests/fast/examples/experimental/test_verifiers_run.py` 移动到新目录并重命名为 `test_run.py`, 去掉 verifiers 冗余前缀, 使测试文件名与 `run.py` 对应, 遵循 `p2p_weight_transfer` 的既有命名先例。
4. 验证 CI 收集结果: PR body 使用 `run_suite.py --hw cpu --suite stage-a-cpu --list-only` 对比, 确认套件收集仅差三个测试名, 且每个仍为隐式 CPU entry, 无需改动 `tests/ci/run_suite.py` 或任意 CI workflow 配置。

关键文件：

- tests/fast/examples/experimental/verifiers/test_run.py（模块 启动器；类别 test；类型 rename-or-move；符号 test_adapter_and_ray_runtime_use_the_same_legacy_flag, _rollout_config, LEGACY_ROLLOUT_ENV）：launcher 测试，从 experimental/ 根目录移入并去掉 verifiers 冗余前缀，重命名后与 run.py 对应，是最能体现本 PR 组织意图的文件。
- tests/fast/examples/experimental/verifiers/test_verifiers_rollout.py（模块 rollout；类别 test；类型 rename-or-move；符号 VerifiersRolloutFn, MilesSGLangTransport, trace_to_samples, _validate_args）：覆盖 VerifiersRolloutFn 及 trace 转换逻辑的测试，从 tests/fast/rollout/ 移入，是本次迁移的核心对象之一。
- tests/fast/examples/experimental/verifiers/test_verifiers_runtime.py（模块 运行时；类别 test；类型 rename-or-move；符号 test_minimal_env_config_uses_the_v1_environment_contract）：验证 verifiers v1 runtime 契约（EnvConfig、Environment、renderer 等）的测试，同样从 tests/fast/rollout/ 移入，使该目录不再混入 example 测试。
- tests/fast/examples/experimental/verifiers/__init__.py（模块 测试包；类别 test；类型 test-coverage）：新增空包标记文件，使新目录成为可导入的测试包，与仓库其他 example 测试目录保持一致。

关键符号：test_adapter_and_ray_runtime_use_the_same_legacy_flag, _rollout_config, test_minimal_env_config_uses_the_v1_environment_contract

关键源码片段

tests/fast/examples/experimental/verifiers/test_run.py

launcher 测试，从 experimental/ 根目录移入并去掉 verifiers 冗余前缀，重命名后与 run.py 对应，是最能体现本 PR 组织意图的文件。

```
import json
import shlex

import pytest
from examples.experimental.verifiers import run
from tests.fast.utils.command_recorder import record_commands

import miles.utils.external_utils.command_utils as U

LEGACY_ROLLOUT_ENV = "MILES_USE_LEGACY_ROLLOUT_V1"

def _rollout_config(submit_command: str) -> tuple[str, dict[str, str]]:
    # 从最终提交给 Ray 的 submit 命令中反向解析出 rollout 函数路径与 runtime 环境变量，
    # 用于断言 adapter 侧与 Ray runtime 侧使用的是同一个 legacy 开关。
    argv = shlex.split(submit_command)
    rollout_fn = argv[argv.index("--rollout-function-path") + 1]
    runtime_env_arg = next(arg for arg in argv if arg.startswith("--runtime-env-json="))
    runtime_env = json.loads(runtime_env_arg.split("=", 1)[1])["env_vars"]
    return rollout_fn, runtime_env
```

```

@pytest.mark.parametrize(
    ("ambient_value", "extra_env_vars", "expected_rollout_fn", "expected_runtime_value"),
    [
        (None, "", "verifiers_rollout.VerifiersRolloutFn", None),
        ("1", "", "verifiers_rollout.generate_rollout", "1"),
        ("0", f"{LEGACY_ROLLOUT_ENV}=1", "verifiers_rollout.generate_rollout", "1"),
        ("1", f"{LEGACY_ROLLOUT_ENV}=0", "verifiers_rollout.VerifiersRolloutFn", "0"),
    ],
)

def test_adapter_and_ray_runtime_use_the_same_legacy_flag(
    monkeypatch,
    tmp_path,
    ambient_value,
    extra_env_vars,
    expected_rollout_fn,
    expected_runtime_value,
):
    # 四种环境变量组合交叉验证：无设置、仅环境变量、环境变量与 extra_env_vars 冲突等场景，
    # 期望最终选中的 rollout 函数与 runtime 环境变量始终一致，避免 adapter 与 runtime
    # 行为分叉。
    commands = record_commands(monkeypatch)
    monkeypatch.setattr(U, "check_has_nvlink", lambda: False)
    monkeypatch.setenv("MILES_SCRIPT_EXTERNAL_RAY", "1")
    monkeypatch.setenv("MILES_SCRIPT_ENABLE_RAY_SUBMIT", "1")
    monkeypatch.setenv("MASTER_ADDR", "127.0.0.1")
    monkeypatch.delenv("RAY_ADDRESS", raising=False)
    monkeypatch.delenv("NCCL_NVLS_ENABLE", raising=False)
    if ambient_value is None:
        monkeypatch.delenv(LEGACY_ROLLOUT_ENV, raising=False)
    else:
        monkeypatch.setenv(LEGACY_ROLLOUT_ENV, ambient_value)

    # 构造最小 ScriptArgs 并执行 run.py 的入口，monkeypatch 会录制最终提交的命令。
    run.execute(
        run.ScriptArgs(
            verifiers_config=str(tmp_path / "verifiers.toml"),
            extra_env_vars=extra_env_vars,
        )
    )

    rollout_fn, runtime_env = _rollout_config(commands[-1])
    assert rollout_fn == expected_rollout_fn
    if expected_runtime_value is None:
        assert LEGACY_ROLLOUT_ENV not in runtime_env
    else:
        assert runtime_env[LEGACY_ROLLOUT_ENV] == expected_runtime_value

```

评论区精华

该 PR 的 review 讨论较少且无争议，核心是 [claude\[bot\]](#) 的自动审核与人工批准。

- [claude\[bot\]](#) 的 LGTM 评论详细验证了 PR 的安全声明：grep 仓库确认无旧路径残留引用，确认三个文件均不使用 `__file__` 避免相对路径假设被破坏，并确认测试全部通过 `examples.experimental.verifiers...` 包路径导入而非路径相对机制，与 PR body 描述完全一致。
- Shi-Dong 给出简洁的 APPROVED (LGTM)，无额外问题。
- 唯一的第一条 issue 评论是作者本人触发的 [@claude review](#)，属于流程性操作。
- [claude\[bot\]](#) 对纯移动安全性的独立验证 (other): 验证结果与 PR body 描述完全一致，无风险，批准。
- 人工审批确认 (other): 批准合并，无遗留问题。

风险与影响

- 风险：技术风险极低。变更纯为文件移动与重命名，无逻辑改动，四个文件的 additions 与 deletions 均为 0。作者已主动验证三点安全性：旧路径无外部引用、无 `__file__` 依赖、均通过包路径导入。潜在残余风险仅在于若仓库外部工具（不在本仓库内的脚本）硬编码旧测试路径，可能在收集时遗漏用例，但这不属于仓库内可见风险。另外，`test_run.py` 重命名后若有人按旧名称 `test_verifiers_run.py` 在 CI 配置中显式引用会失效，但 PR 已确认 `tests/ci` 无此类引用。
- 影响：影响范围限于测试代码组织层面：`tests/fast/rollout/` 目录的职责回归纯粹（仅覆盖 `miles` 自有 `rollout` 代码），`verifiers example` 的测试集中在独立目录下，降低后续维护者寻找用例的心智负担。对用户功能无影响，对 CI 收集结果仅三个测试名的路径变化，无执行行为变化。对团队而言，确立了“example 测试按 example 路径归档”的组织惯例，为后续 `swe_agent_harbor`、`p2p_weight_transfer` 之外的 example 测试提供先例。
- 风险标记：纯测试移动无逻辑变更，旧路径无外部引用已确认，目录镜像原则可复用

关联脉络

- PR #2558（作者在 PR body 中提及的前置 PR，已合并）：本 PR 基于 #2558 的移动成果 rebase 到 main，将相邻 suites 移入 `tests/fast/` 下的对应目录，本 PR 是其后续整理。
- PR #2577 fix: make the Verifiers E2E self-contained: 后续对 `verifiers` 测试的配套修复，涉及 `examples/experimental/verifiers/verifiers_rollout.py` 与 `tests/fast/rollout/test_verifiers_rollout.py`（在 2561 合并前路径），与本 PR 属同一 `verifiers` 测试工作线。
- PR #2560（PR body 中提到的后续工作项）：PR body 明确声明 `verifiers e2e` 固定 legacy rollout 路径、默认 `VerifiersRolloutFn` 无端到端覆盖的问题属于 #2560，与本 PR 同源但刻意拆分，避免在纯移动 PR 中混入行为变更。