

PR #2485 完整报告

radixark/miles

clean up fully async example and mv to examples/infra_features

合并时间: 2026-08-13 10:58

原文链接: <http://prhub.com.cn/radixark/miles/pull/2485>

执行摘要

- 一句话: fully_async 示例迁至 infra_features 并删除 shell 启动器
- 推荐动作: 值得快速阅读: 它展示了 " 示例目录迁移 " 这类看似机械的改动如何被 dotted module path 加载机制放大为跨模块联动, 以及如何通过删除重复 README、依赖自动文档生成来降低长期维护成本。对示例和文档维护者尤其有参考价值, 可关注其 " 删除 shell 启动器 " 与 " 用链接替代重复内容 " 两个决策。

功能与动机

PR body 明确说明: fully-async 属于基础设施功能, 应当归入 examples/infra_features/, 与 random_async、true_on_policy 等并列, random_async 的 README 也自称是它的 "minimal sibling"。shell 启动器被删除而非移植, 因为它没有展示两个 Python 启动器未覆盖的能力——README 点名的三项变更 (train_async.py、MILES_EXPERIMENTAL_ROLLOUT_REFACTOR=1、--fully-async) 都已存在于 run_qwen3_5_4b_fully_async_eval.py 中, 其唯一特有的注释态 --max-weight-staleness 参数在文档中已有解释。

实现拆解

整个变更按以下 5 步展开:

1. 目录搬迁 (git rename) : 将 examples/fully_async/ 整体移动到 examples/infra_features/fully_async/, 保留 init.py、external_eval_fn.py、run_qwen3_5_4b_fully_async_eval.py、run_qwen3_30b_a3b_fully_async.py 四个文件, 内容零改动, 仅路径变化。
2. dotted module path 联动更新: external_eval_fn 是通过模块路径而非文件路径加载的 (checkpoint_eval 契约), 因此迁移后必须同步所有引用点: run_qwen3_5_4b_fully_async_eval.py 的 --eval-function-path 更新为 examples.infra_features.fully_async.external_eval_fn.ExternalSglangEvalFn; tests/fast/rollout/test_checkpoint_eval.py 的 importlib.import_module 调用同步更新; tests/e2e/megatron/test_qwen3_4b_fully_async_eval.py 的 --eval-function-path 同步更新; miles/rollout/checkpoint_eval.py 的 docstring 引用同步更新。这是本次迁移 " 非纯机械 " 的核心原因。

3. 删除 shell 启动器：删除 `run-qwen3-4b-fully_async.sh` (143 行) 及其快照 `tests/snapshots/launch_scripts/sh/examples/fully_async/run-qwen3-4b-fully_async.sh.txt` (192 行)；`tests/fast/test_ray_launcher_unbuffering.py` 作为 ray launcher 计数守卫，计数因此减 1。
4. README 与文档去重：原 `examples/fully_async/README.md` (51 行) 约 30 行与两份文档重复，重写为链接导向的精简 README，置顶 <https://miles.radixark.com/docs/> 链接；`docs/examples/fully-async.md` 删除，由自动生成流程产出 `docs/examples/infra-features/fully-async.md`；同步更新 `docs/docs.json` 导航、`docs/user-guide/fully-async.md`、`examples/README.md`、`random_async/README.md`、`docs/models/glm/glm5-2.md` 等路径引用，并补上 README 文件清单中缺失的 `run_qwen3_30b_a3b_fully_async.py`。
5. 测试验证：执行 `pytest tests/fast/rollout/test_checkpoint_eval.py tests/manual/launch_scripts tests/fast/launch_scripts tests/fast/test_ray_launcher_unbuffering.py`，共 586 个测试通过；`pre-commit run --all-files` 干净；最后一个 commit 为 merge main，解决与自动文档生成流程的冲突。

关键文件：

- `examples/infra_features/fully_async/run_qwen3_5_4b_fully_async_eval.py` (模块 示例脚本；类别 source；类型 rename-or-move；符号 execute)：fully_async 示例的主入口 launcher，迁移后唯一的实质代码改动点：`--eval-function-path` 从 `examples.fully_async` 更新为 `examples.infra_features.fully_async`，是 dotted module path 联动的关键证据。
- `examples/fully_async/run-qwen3-4b-fully_async.sh` (模块 启动脚本；类别 other；类型 deletion)：目录中唯一的 shell 启动器，本次被删除而非移植，体现示例维护策略：Python launcher 已覆盖全部关键配置，shell 脚本仅剩注释态的 `--max-weight-staleness`。
- `miles/rollout/checkpoint_eval.py` (模块 评估后端；类别 source；类型 core-logic)：CheckpointEvalFn 契约所在的核心库文件，虽然只改 docstring，但它是外部评估函数加载机制的源头，迁移后引用路径必须与之一致。
- `examples/infra_features/fully_async/external_eval_fn.py` (模块 评估后端；类别 source；类型 rename-or-move；符号 ExternalSglangEvalFn.init)：参考 CheckpointEvalFn 实现，负责自建或挂接外部 sglang server；文件内容零改动，但它是 dotted module path 机制的载体。
- `tests/fast/rollout/test_checkpoint_eval.py` (模块 单元测试；类别 test；类型 test-coverage；符号 external_fn_env)：CI 中直接按 dotted module path 加载示例评估函数的测试，迁移后 import 路径同步更新，是验证迁移正确性的关键测试。
- `examples/infra_features/fully_async/README.md` (模块 示例文档；类别 docs；类型 documentation)：新增的精简 README，用链接替代与文档页重复的约 30 行内容，置顶官方文档 URL，体现防文档漂移的维护策略。
- `tests/snapshots/launch_scripts/sh/examples/fully_async/run-qwen3-4b-fully_async.sh.txt` (模块 快照测试；类别 docs；类型 deletion)：被删 shell 启动器的命令展开快照，与脚本一并移除，避免快照测试与 launch_scripts 目录状态不一致。
- `docs/examples/fully-async.md` (模块 文档页面；类别 docs；类型 deletion)：旧的手动文档页被删除，改由自动生成流程产出 `docs/examples/infra-features/fully-async.md`，反映

文档维护方式的转变。

关键符号: `execute`, `ExternalSglangEvalFn.init`, `external_fn_env`

关键源码片段

[examples/infra_features/fully_async/run_qwen3_5_4b_fully_async_eval.py](#)

`fully_async` 示例的主入口 `launcher`, 迁移后唯一的实质代码改动点: `--eval-function-path` 从 `examples.fully_async` 更新为 `examples.infra_features.fully_async`, 是 dotted module path 联动的关键证据。

迁移后的 `launcher` 中, `execute()` 依据 `eval` 后端选择不同路径。

关键点: 外部评估函数通过 dotted module path 加载, 而不是文件路径,

所以目录搬迁必须同步这里, 否则 `importlib` 找不到模块。

```
eval_args = (
    "--eval-interval 5 "
    f"--eval-prompt-data aime {args.data_dir}/aime-2024/aime-2024.jsonl "
    "--n-samples-per-eval-prompt 8 "
    "--eval-max-response-len 16384 "
    "--eval-top-p 1 "
    "--eval-hf-dir /dev/shm/miles_eval_hf "
    "--eval-keep-snapshots 2 "
)

if args.eval_backend == "fleet":
    # fleet 模式: 训练作业内分配 1 张 GPU 作为专用 eval 引擎。
    eval_args += "--eval-num-gpus 1 --eval-num-gpus-per-engine 1 "
else:
    # external 模式: 把评估函数注册为外部后端,
    # 函数会在最后一张空闲 GPU 上自建 sglang server, 独立于 Ray 集群。
    eval_args += (
        "--eval-function-path "
        "examples.infra_features.fully_async.external_eval_fn.ExternalSglangEvalFn "
    )
    eval_env = {"MILES_EXTERNAL_EVAL_GPUS": str(args.num_gpus_per_node - 1)}
```

[examples/infra_features/fully_async/external_eval_fn.py](#)

参考 `CheckpointEvalFn` 实现, 负责自建或挂接外部 `sglang server`; 文件内容零改动, 但它是 dotted module path 机制的载体。

`external_eval_fn.py` 的构造入口: 准备一个独立 `sglang` 后端。

该文件被移动后代码本身零改动, 只有模块导入路径随之更新。

```
class ExternalSglangEvalFn(CheckpointEvalFn):
    """启动 (或挂接) 独立 sglang server, 并在快照上执行评估。"""

    def __init__(self, input: RolloutFnConstructorInput):
        args = input.args
        url = os.environ.get("MILES_EXTERNAL_EVAL_URL") # 挂接已有服务
        gpus = os.environ.get("MILES_EXTERNAL_EVAL_GPUS") # 或自行拉起服务
```

```

self._proc: subprocess.Popen | None = None
if url is None:
    if gpus is None:
        raise ValueError(
            "ExternalSglangEvalFn needs a backend: set "
            "MILES_EXTERNAL_EVAL_URL (attach) or "
            "MILES_EXTERNAL_EVAL_GPUS (launch our own, e.g. '6,7')"
        )
    # launch 模式: 在指定 GPU 上启动 sglang 服务, tp 数等于 GPU 数。
    # 服务进程由 self._proc 持有, 后续 evaluate_checkpoint 会
    # 按权重版本挂载 checkpoint 目录并运行标准 eval 数据集。
    port = int(os.environ.get("MILES_EXTERNAL_EVAL_PORT", "31000"))
    num_gpus = len(gpus.split(","))
    cmd = [
        sys.executable, "-m", "sglang.launch_server",
        "--model-path", args.hf_checkpoint,
        "--tp", str(num_gpus),
        "--host", "127.0.0.1",
        "--port", str(port),
        "--mem-fraction-static", "0.8",
        "--trust-remote-code",
    ]
    self._proc = subprocess.Popen(cmd) # 实际还会附加 SERVER_ARGS 环境变量参数

```

tests/fast/rollout/test_checkpoint_eval.py

CI 中直接按 dotted module path 加载示例评估函数的测试, 迁移后 import 路径同步更新, 是验证迁移正确性的关键测试。

```

# tests/fast/rollout/test_checkpoint_eval.py 中的 fixture:
# 通过 importlib 按 dotted module path 加载示例评估函数,
# 因此目录迁移后这里必须同步更新, 否则 CI 会直接失败。
@pytest.fixture
def external_fn_env(monkeypatch):
    import importlib

    mod = importlib.import_module(
        "examples.infra_features.fully_async.external_eval_fn"
    )
    # 用 monkeypatch 替换网络调用, 把外部 sglang 服务隔离为内存假对象,
    # 使测试不依赖真实 GPU 与服务器。
    ...

```

评论区精华

核心讨论来自 nblintao 的 review 与 issue 评论。在 issue 评论中他指出: "probably need to rebase and fix conflicts. i added auto doc webpage generation. now docs/examples fully mirrors /examples"——即仓库已上线 docs/examples 自动镜像 /examples 的生成流程, 本 PR 原本手动维护 docs/examples/fully-async.md 的做法已不再需要。review 状态为

APPROVED, 但要求 rebase。最终作者通过 merge main (提交 1ee005c) 解决冲突, 并删除了手动维护的文档页, 改为依赖自动生成。另外 PR body 中详细论证了删除 shell 启动器的取舍: 目录中三个启动器只有它是 shell, Python 启动器已覆盖 README 点名的全部配置差异, 唯一独特内容 (被注释的 `--max-weight-staleness`) 文档中已有解释, 因此不移植。

- docs/examples 自动生成与手动维护冲突 (documentation): 作者通过 merge main (提交 1ee005c) 解决冲突, 删除手动维护的 docs/examples/fully-async.md, 改由自动生成流程产出新页面。

风险与影响

- 风险: 主要风险集中在 dotted module path 的联动更新上: external_eval_fn 通过 importlib 按模块路径加载, 若迁移后残留任何旧路径引用 (文档示例、用户自建脚本), 会在运行时直接抛 ImportError。CI 中 tests/fast/rollout/test_checkpoint_eval.py 的 external_fn_env fixture 覆盖了该路径, e2e 测试也做了同步更新, 但仓库外的存量脚本不受此次测试保护。其次, 删除 run-qwen3-4b-fully_async.sh 对依赖旧命令的用户是行为破坏, 快照测试已同步删除, 但需确认无其他文档或脚本引用该命令。最后, docs/examples 的自动生成流程若与本 PR 的手动文档改动再次冲突, rebase 可能反复; 合并完成后 docs/examples/fully-async.md 已删除, 该风险已消除。
- 影响: 影响范围集中在示例目录结构与文档导航: examples/fully_async 路径整体失效, examples/infra_features/fully_async 成为新位置; 文档站 docs/examples 下的页面路径随之变化, docs/docs.json 导航已同步。对用户而言, 旧路径的启动命令和示例链接需要更新; 对团队而言, 示例归类与文档维护方式更一致, 减少了 README 与文档双份维护带来的漂移。对训练核心逻辑无影响 (miles/rollout/checkpoint_eval.py 仅 docstring 变化), 整体影响程度中等偏小。
- 风险标记: dotted module path 断裂风险, shell 启动器删除为 breaking change, 自动文档生成流程冲突

关联脉络

- PR #2356 Replace all the .sh launch scripts with .py launch script: 仓库正在把 .sh 启动脚本全面替换为 Python 启动器, 本 PR 删除 fully_async 目录下唯一的 shell 启动器, 与这一迁移方向一致。
- PR #2482 examples: fix READMEs — dead links, broken list structure, missing entries: 同为 examples 目录的文档维护工作, 本 PR 也修正了 examples/README.md 中 fully_async 的目录引用与文件清单遗漏。