

PR #2522 完整报告

radixark/miles

Make the class-based rollout the default and convert legacy path to env var gated

合并时间: 2026-08-14 13:05

原文链接: <http://prhub.com.cn/radixark/miles/pull/2522>

执行摘要

- 一句话: class-based rollout 成为默认, 旧路径改为环境变量门控
- 推荐动作: 值得精读。该 PR 展示了 "实验变产品" 的典型演进: 极性翻转 + fail-fast + adapter 兼容三层保护, 是基础设施迁移的教科书式操作。重点关注 miles/utils/arguments.py 的 v1-only 校验模式、inference_rollout_common.py 的 lifecycle 收尾设计, 以及 tests/fast/examples/experimental/test_verifiers_run.py 的环境一致性参数化测试。后续可跟进 v1 模块移除的 PR。

功能与动机

PR body 明确指出: `MILES_EXPERIMENTAL_ROLLOUT_REFACTOR` 自 1 月起就门控着 class-based rollout 路径 (#484/#491), 而此后新增的 fully-async、snapshot eval、agentic tool calls、nemo-gym、openenv、verifiers 全都依赖它, "The experiment is the product". 因此需要翻转极性, 让 `InferenceRolloutFn` 成为默认, v1 仅通过 `MILES_USE_LEGACY_ROLLOUT_V1=1` 保留, 为后续彻底移除 v1 模块铺路。

实现拆解

1. 极性翻转: 环境开关从 opt-in 变为 opt-out

miles/utils/environs.py 用 `use_legacy_rollout_v1()` 取代 `enable_experimental_rollout_refactor()`, 默认读取 `MILES_USE_LEGACY_ROLLOUT_V1` 且默认为 0 (即默认走 class-based 路径)。所有消费点同步翻转判断: miles/utils/arguments.py 的 `resolve_rollout_function_paths` 默认返回 `miles.rollout.inference_rollout.inference_rollout_common.InferenceRolloutFn`; miles/ray/rollout/rollout_manager.py 将 `use_experimental_refactor` 属性改为 `use_legacy_rollout_v1`, train/eval 调用点取反; `fully_async_rollout.py`、`inference_rollout_train.py`、`checkpoint_eval` 桩也一并更新。

2. v1-only 特性 fail-fast

`_resolve_rollout_functions` 和 `miles_validate_args` 新增两道拦截: `--mask-offpolicy-in-partial-rollout` (class-based 路径 resume 后不重扩展 loss mask) 和 `--opd-log-prob-top-k` 配合 student-side 策略 (需要 v1 generate 才产生的 `opd_student_top_logprobs`) 在参数校验阶段直接 `raise ValueError`, 提示设置 `MILES_USE_LEGACY_ROLLOUT_V1=1`, 而不是在 run 中途深埋报错。

3. 补齐默认路径的观测缺口 (review 驱动)

miles/rollout/inference_rollout/inference_rollout_common.py 的 generate_and_rm 引入 TrajectoryLifecycle 事件 (attempt_start / gen_start / attempt_end), 用 try/finally 保证异常与 sibling 取消时也收尾; 同时支持 per-sample generate_function_path 覆盖。

miles/rollout/sglang_rollout.py 的私有函数 _sample_text_preview / _reward_log_summary / _len_or_value 被提升为 generate_utils/sample_utils.py 的共享 sample_text_preview / reward_log_summary, 并在 class-based 与 fully-async 驱动中复用, 避免长上下文与 OPD 大 reward 刷爆 Ray 日志。

4. 适配层与示例修复

load_rollout_function / load_generate_function 的 LegacyRolloutFnAdapter / LegacyGenerateFnAdapter 保持不变, 保证旧自定义函数在默认路径下继续工作; examples/experimental/verifiers/run.py 改为根据同一份 effective env (含 resolve_extra_env_vars 合并后的结果) 选择 adapter 与 Ray runtime env, 消除 launcher 读 os.environ 与 runtime env 不一致的隐患; miles/utils/external_utils/command_utils.py 抽出 resolve_extra_env_vars 公共函数。

5. 清理 opt-in 与配套

14 个 launcher/example 删除 =1; 30 个 e2e 测试去掉 opt-in; tests/conftest.py 的 autouse fixture 从 enable 改为清除 ambient MILES_USE_LEGACY_ROLLOUT_V1 (幂等防护); 6 个 LoRA snapshot 从记录 runtime env 移除该变量、2 个 OPD shell snapshot 增加 pin; tests/fast/examples/experimental/test_verifiers_run.py 新增参数化测试覆盖 ambient / extra_env_vars 四种组合; generate-endpoint.md、customization.md 等 10 个文档页面重写, 将默认路径表述对齐为 class-based。

关键文件:

- miles/utils/arguments.py (模块 参数校验; 类别 source; 类型 dependency-wiring; 符号 resolve_rollout_function_paths, _resolve_rollout_functions, miles_validate_args): 默认路径选择与 v1-only 特性 fail-fast 的核心决策点, 直接决定 rollout 函数解析与参数校验行为
- miles/utils/envIRON.py (模块 环境开关; 类别 source; 类型 core-logic; 符号 use_legacy_rollout_v1): 环境开关极性翻转的源头, 旧 flag 被移除、新 flag 语义反转为 opt-out
- miles/rollout/inference_rollout/inference_rollout_common.py (模块 轨迹引擎; 类别 source; 类型 core-logic; 符号 generate_and_rm): class-based 路径成为默认后补齐 dashboard 生命周期事件与 per-sample generate override, 是 review 驱动的关键补强
- miles/ray/rollout/rollout_manager.py (模块 调度器; 类别 source; 类型 dependency-wiring; 符号 RolloutManager): RolloutManager 的 train/eval/loader 三处调用点随开关翻转, 是调度侧的主消费方
- miles/rollout/sglang_rollout.py (模块 旧版路径; 类别 source; 类型 core-logic; 符号 generate_rollout): v1 路径随翻转被降级为 legacy, 本地日志辅助函数迁移到共享 sample_utils

- miles/rollout/generate_utils/sample_utils.py (模块 采样工具; 类别 source; 类型 core-logic; 符号 sample_text_preview, reward_log_summary, _len_or_value) : 样本预览与 reward 摘要从 v1 专用提升为两条路径共享的工具, 防止 class-based 默认路径日志轰炸
- tests/fast/rollout/inference_rollout/test_lifecycle_attempt.py (模块 生命周期; 类别 test; 类型 test-coverage; 符号 RecordingSink, test_attempt_ends_once_after_success, test_attempt_ends_once_after_abort, test_attempt_ends_when_generate_raises) : 新增生命周期事件顺序测试, 覆盖成功 /abort/ 异常 /sibling 取消四种路径, 锁定 try/finally 收尾行为
- tests/fast/examples/experimental/test_verifiers_run.py (模块 验证器; 类别 test; 类型 test-coverage; 符号 test_adapter_and_ray_runtime_use_the_same_legacy_flag) : 新增参数化测试验证 launcher 的 adapter 选择与 Ray runtime env 使用同一份 legacy flag, 锁定 #2528 修复
- examples/experimental/verifiers/run.py (模块 验证器; 类别 source; 类型 core-logic; 符号 execute) : 修复 adapter 选择读 os.environ 与 Ray runtime env 不一致的隐患, 改为基于合并后的 effective env 决策
- miles/utils/external_utils/command_utils.py (模块 启动命令; 类别 source; 类型 core-logic; 符号 resolve_extra_env_vars) : 抽出 resolve_extra_env_vars 供 verifiers 示例复用, 避免启动命令与 runtime env 拼接逻辑分叉

关键符号: use_legacy_rollout_v1, resolve_rollout_function_paths, _resolve_rollout_functions, miles_validate_args, generate_and_rm, sample_text_preview, reward_log_summary, resolve_extra_env_vars, execute

关键源码片段

miles/utils/arguments.py

默认路径选择与 v1-only 特性 fail-fast 的核心决策点, 直接决定 rollout 函数解析与参数校验行为

miles/utils/arguments.py — 默认路径选择与 v1-only 特性 fail-fast

```
def resolve_rollout_function_paths(args) -> tuple[str, str]:
    """返回 (rollout, eval) 函数路径。默认 class-based, legacy 需显式 opt-in."""
    if use_legacy_rollout_v1():
        # v1 旧路径: `MILES_USE_LEGACY_ROLLOUT_V1=1` 时保留的 sglang_rollout
        standard_path = "miles.rollout.sglang_rollout.generate_rollout"
    else:
        # 新默认: class-based `InferenceRolloutFn`
        standard_path = "miles.rollout.inference_rollout.inference_rollout_common.
            InferenceRolloutFn"
    rollout_path = args.rollout_function_path or standard_path
    if args.fully_async:
        # 完全异步只存在于 class-based 世界
        rollout_path = "miles.rollout.fully_async_rollout.FullyAsyncRolloutFn"
    ...
```

```

def _resolve_rollout_functions(args) -> None:
    # v1-only 特性在新默认路径上 fail-fast, 而不是在 run 中途深埋报错;
    # 提示用户设置 `MILES_USE_LEGACY_ROLLOUT_V1=1` 回退旧路径
    if args.partial_rollout and args.mask_offpolicy_in_partial_rollout and not use_legacy_rollout_v1():
        raise ValueError(
            "--mask-offpolicy-in-partial-rollout does not re-extend the loss mask on the "
            "class-based rollout path yet; set MILES_USE_LEGACY_ROLLOUT_V1=1"
        )
    if args.fully_async:
        assert (
            not use_legacy_rollout_v1()
        ), "--fully-async needs the class-based rollout API; unset MILES_USE_LEGACY_ROLLOUT_V1"
    ...

```

miles/utils/envIRON.py

环境开关极性翻转的源头, 旧 flag 被移除、新 flag 语义反转为 opt-out

```

# miles/utils/envIRON.py — 环境开关极性翻转的核心
import os

```

```

_printed_legacy_rollout_v1 = False

```

```

def use_legacy_rollout_v1() -> bool:
    # 默认返回 False: 所有任务默认走 class-based `InferenceRolloutFn`;
    # 只有显式设置 `MILES_USE_LEGACY_ROLLOUT_V1=1` 才回退到 v1 `sglang_rollout` 路径
    result = bool(int(os.environ.get("MILES_USE_LEGACY_ROLLOUT_V1", "0")))

    global _printed_legacy_rollout_v1
    if result and not _printed_legacy_rollout_v1:
        # 只在首次命中时打一条提示, 避免每个进程都刷屏
        print("MILES_USE_LEGACY_ROLLOUT_V1=1 is enabled: using the deprecated v1 rollout path")
        _printed_legacy_rollout_v1 = True

    return result

```

miles/rollout/inference_rollout/inference_rollout_common.py

class-based 路径成为默认后补齐 dashboard 生命周期事件与 per-sample generate override, 是 review 驱动的关键补强

```

# miles/rollout/inference_rollout/inference_rollout_common.py — 新默认路径的生命周期收尾
async def generate_and_rm(
    state: GenerateState,
    sample: Sample | list[Sample],
    sampling_params: dict[str, Any],
    evaluation: bool = False,

```

```

) -> Sample | list[Sample]:
    args = state.args

    # partial rollout 时, 按 loss mask 屏蔽上一轮 off-policy 片段
    if args.partial_rollout and args.mask_offpolicy_in_partial_rollout and sample.response_length
    > 0:
        sample.loss_mask = [0] * sample.response_length

    # 已有完整回复的样本直接返回, 不再重复生成
    if sample.status in (Sample.Status.COMPLETED, Sample.Status.TRUNCATED):
        assert sample.response is not None
        if not args.group_rm:
            assert sample.reward is not None
        return sample

    # dashboard 生命周期探针: 信号量等待即“排队”时间;
    # attempt_start 在排队前发出, attempt_end 在 reward 计算前发出
    sink = None if evaluation else TrajectoryLifecycle().sink
    if sink is not None:
        sink.attempt_start(sample)

    try:
        async with state.generate_fn_semaphore:
            if state.aborted:
                # 已终止的 rollout 不再触发 generate
                sample.status = Sample.Status.ABORTED
                return sample

            if sink is not None:
                sink.gen_start(sample)
            # per-sample override: eval dataset 可自行指定 generate function
            generate_fn = load_generate_function(sample.generate_function_path) or state.
            generate_function
            output = await generate_fn(
                GenerateFnInput(
                    state=state,
                    sample=sample,
                    sampling_params=deepcopy(sampling_params),
                    evaluation=evaluation,
                )
            )
            sample = output.samples
    finally:
        # 无论成功、抛异常还是被 sibling 取消, 都必须 attempt_end,
        # 否则 dashboard 会认为该轨迹一直 running
        if sink is not None:
            sink.attempt_end(sample)
    ...

```

评论区精华

guapisolo 在 review 中提出 4 条 [P2] 意见，全部在合入前解决：

1. 新默认路径缺失 TrajectoryLifecycle 事件 (design, 涉及 `inference_rollout_common.generate_and_rm`) : "The legacy path emits `attempt_start`, `gen_start`, and `attempt_end` around the semaphore/generate call, and the dashboard consumes those events for its per-sample queue/generation timeline. Consequently, `--use-miles-dashboard` jobs on the new default path have an empty trajectory stream even though generation succeeds." 已通过 `a6e305d` 提交在 class-based 路径补齐生命周期探针。
 2. 日志截断与 reward 摘要缺失 (performance) : "inference_rollout_train.py interpolates the complete prompt, response, and raw reward... every rollout can now send very large messages through Ray log forwarding and storage." 已通过 `6c21659` 提交将 v1 的预览 / 摘要逻辑共享到 `sample_utils.py`。
 3. verifiers launcher 与 Ray runtime env 不一致 (correctness) : "That can select `VerifiersRolloutFn` while `RolloutManager` uses the v1 ABI, raising `TypeError` on the first rollout. I put the root-cause fix in #2528." 已通过 `75050be` 提交合入 #2528 的修复，并由新增参数化测试锁定。
 4. lifecycle 在 failure/cancellation 未 close (correctness) : "Both paths skip the normal-path `attempt_end()`... the dashboard can leave an already-finished trajectory in the running state indefinitely." 已通过 `e2b289a` 提交 (#2534) 用 `try/finally` 保证 `attempt_end` 兜底。
- 新默认路径缺失 TrajectoryLifecycle 事件导致 dashboard 轨迹流为空 (design): 在 class-based 路径补上相同的生命周期插桩，同时避免专用生成器产生重复 per-turn span。已通过 `a6e305d` 提交实现，并由新增 `test_lifecycle_attempt.py` 锁定。
 - 新默认路径的 first/finish 日志未截断，长上下文与 OPD 大 reward 会刷爆 Ray 日志 (performance): 将 legacy 的预览 / 摘要格式提升为 `sample_utils.py` 共享函数并应用到 class-based 与 fully-async 驱动。已通过 `6c21659` 提交解决。
 - verifiers launcher 的 `os.environ` 与 Ray runtime env 不一致会选中错误 adapter 导致 `TypeError` (correctness): 根因修复在 #2528: adapter 选择与传播的 flag 使用同一份 effective env, CLI 覆盖优先于 ambient 值。已通过 `75050be` 提交合入本分支，并由新增参数化测试锁定。
 - generate 抛异常或 sibling 取消时 lifecycle attempt 未关闭 (correctness): 从异常 / 取消清理路径统一发出 `attempt_end()` 并保留原始异常。已通过 `e2b289a` 提交 (#2534) 用 `try/finally` 实现。

风险与影响

- 风险：
 1. 约 30 个 e2e 测试静默切换路径：这些纯 GRPO/SFT 任务原先走 v1，现在默认走 class-based。PR body 列出了三处刻意差异——per-group task 异常改为记录并重提交（不再崩溃）、abort fan-out 失败变为 fatal、eval prompt cache 从进程级改为实例级

——首轮 CI 需要观察这些行为变化是否影响训练语义。

2. 环境变量极性反转的兼容性：旧的 MILES_EXPERIMENTAL_ROLLOUT_REFACTOR=1 已被删除，任何仍在设置该变量的脚本 / 文档会静默失效；反之 ambient MILES_USE_LEGACY_ROLLOUT_V1=1 会污染本地开发与 snapshot 录制，已通过 tests/confest.py 的 CLEARED_ENV 防护，但外部用户的遗留 shell 环境仍可能误切换到 v1。
 3. 可观测性盲区：dashboard 的轨迹事件只由 class-based 路径发出；若用户 opt-in v1，dashboard 轨迹流将为空（v1 侧未补 sink）。这是门控翻转带来的新的功能不对称。
 4. v1-only 特性未被全覆盖：audio_data payload、per-eval-dataset custom_generate_function_path、kimi-style call_processor 多模态 prompt 等 gap 无 in-repo 消费者，未做 fail-fast，遇到才会暴露。- 影响：用户与任务：所有未显式设置 legacy 变量的 rollout 任务默认切换到 InferenceRolloutFn，但 Legacy adapter 保证旧 --rollout-function-path / --custom-generate-function-path 调用约定不变，sft_rollout、random_async、multi-LoRA 注入函数不受影响。系统与团队：这是 v1 移除计划的前置里程碑，MILES_USE_LEGACY_ROLLOUT_V1 为迁移窗口期的安全网；CI 中 test_qwen3_0.6B_verifiers 成为唯一显式 legacy pin，维持 v1 覆盖直到路径删除。范围：84 个文件、443 个 launch-script snapshot 测试全绿，10 个文档页面同步更新，对团队文档维护和 example 演进影响较大。
- 风险标记：默认路径切换，84 文件联动，环境变量极性反转，30 个 e2e 静默改道，可观测性行为差异

关联脉络

- PR #2528 fix(verifiers): align adapter with Ray runtime env: 本 PR 分支合入的修复，解决 verifiers launcher 的 adapter 选择与 Ray runtime env 使用不同环境变量导致 TypeError 的问题
- PR #2534 fix(rollout): finalize lifecycle attempts on failure: 本 PR 分支合入的修复，用 try/finally 保证 generate 失败或取消时 attempt_end 仍触发
- PR #2531 fix(flops): stop assuming every HF config has intermediate_size: 同属 rollout/FSDP 训练链路收敛，与 class-based 默认路径切换共同推进 megatron/FSDP 路径的稳定性
- PR #2536 Carry the rollout id on both agentic paths: 同为 rollout 路径收敛（agentic v1/v2 rollout_id 一致性），与本 PR 的默认路径翻转互为铺垫
- PR #2485 clean up fully async example and mv to examples/infra_features: fully_async 示例重构依赖 class-based 成为默认，本 PR 后 fully_async 不再需要 opt-in 环境变量
- PR #2484 docs: add disaggregated RL rollout guide: 分离式 rollout 文档基于 class-based 路径编写，与本 PR 的文档默认路径对齐工作直接相关
- PR #2498 fix: require shared rewards within rollouts: rollout 数据转换的奖励一致性修复，同属 rollout 默认路径的服务质量保障
- PR #2219 [feat] Add training log-prob reuse to skip the redundant forward-only pass: megatron 训练侧 log-prob 复用改动，与本 PR 共同构成 rollout+ 训练主路径的演

进