PR #1921 完整报告

radixark/miles

Add NeMo-Gym integration: mini_swe_agent_2 via the agent function

合并时间：2026-07-30 05:36

原文链接：http://prhub.com.cn/radixark/miles/pull/1921

执行摘要

- 一句话：新增 NeMo-Gym 集成示例，agent 函数层对接 mini_swe_agent_2
- 推荐动作：值得精读。三个设计决策有借鉴价值：① agent function 层连接器的失败语义设计——传输失败返回 None 保留 session 成样、reward 兜底 0.0，未触达模型的 episode 经 check_no_aborted 整组丢弃，把 " 环境故障 " 与 " 模型能力不足 " 两类情况做了干净区隔；② 无 GPU 双层扫描验证方法论——golden 扫描以 reward 1.0 为硬门槛验证沙箱 + 镜像 + SWE-bench harness 链路，API-policy 扫描验证 policy_base_url 覆盖的端到端往返，两关都过了才上 GPU 训练，成本极低；③ 上游优先集成策略——把 per-request policy 覆盖做成上游 NeMo-Gym PR（#2166）而非 miles 侧 hack，README 明确指向 PR 分支并说明合并后如何切回上游。若团队计划继续扩展外部环境生态，此 PR 可作为第三个范式实例的模板。

功能与动机

PR body 明确了两层动机：其一，docs/user-guide/environments.md 将 Miles 定位为对环境来源中立（agnostic），已有 Harbor、OpenEnv 等外部生态连接器，本 PR 把 NVIDIA 的 NeMo-Gym 生态以同一 agent function 层形态补进这张表；其二，它取代 #1918 移除的 fork-based generate-function 集成——旧方案对消息文本重新分词且依赖两个个人 fork submodule，新方案完全走上游 NeMo-Gym（无 submodule、无 fork），并经 session server 无损记录 token ids + logprobs + loss masks、绝不重新分词。训练侧需要每个 episode 一条独立 OpenAI 兼容 URL 以无损记录策略调用，这直接催生了上游 NVIDIA-NeMo/Gym#2166 的 per-request policy_base_url 覆盖提案。

实现拆解

1. 连接器（agent function 层）：新增 examples/experimental/nemo-gym/nemogym_agent_function.py。run() 对齐 miles 的 --custom-agent-function-path 约定，每个训练样本调用一次：把 metadata 中的 SWE-bench 实例字段（instance_id、repo、base_commit、problem_statement、subset、split 等）展开到 /run body 顶层（服务端用 body.model_dump() 当实例字典选镜像、跑评测），采样参数经 build_responses_create_params() 映射为 Responses-API 的 temperature / top_p / max_output_tokens（未设置的键省略），policy_base_url 由 _resolve_session_url() 从 session base_url 拼 /v1 并按 MILES_ROUTER_EXTERNAL_HOST 改写 host。post_json() 对传输错误做 3 次随机退避重试，整体由 asyncio.wait_for 按

NEMO_GYM_RUN_TIMEOUT（默认 3600 秒）兜底。文件刻意不 import miles（避免拉入 torch），保证纯 CPU 机器可加载、可离线测试。失败语义：传输失败返回 None，已记录 session 保留成样、reward 兜底 0.0；未触达模型的 episode 无 session 记录，被 generate 层标为 ABORTED，由 check_no_aborted 动态采样过滤器丢弃整组样本。

2. 奖励链路：新增 nemogym_generate.py，reward_func() 从 sample.metadata["reward"] 读取 NeMo-Gym 在 episode 内用官方 SWE-bench harness 预计算的评分，支持单样本与列表两种调用形态，缺失回退 0.0（对应 agent 函数返回 None 的场景）。
3. 数据准备：新增 download_and_process_data.py，支持 HuggingFace dataset（如 princeton-nlp/SWE-bench_Verified，split=test）与本地 JSONL 两种输入，输出 miles prompt 数据 {"prompt": problem_statement, "metadata": { 完整实例 + subset/split}}；subset 控制 NeMo-Gym 按任务选镜像与评测（gym=SWE-Gym 镜像，verified= 官方 SWE-bench 镜像）。
4. 训练启动器：新增 run.py（review 中从 .sh 改为 .py）。ScriptArgs 承载 smoke 规模参数，execute() 拼装 checkpoint / rollout / GRPO / optimizer / 并行 / sglang 全套参数，核心是 agent 链四件套：custom-generate-function-path（miles.rollout.generate_hub.agentic_tool_call.generate）、custom-agent-function-path（nemogym_agent_function.run）、custom-rm-path（nemogym_generate.reward_func）、dynamic-sampling-filter-path（check_no_aborted），并启用 session server（绑 0.0.0.0:30000 供 NeMo-Gym 宿主机从任意网卡回连）。环境变量必须设 MILES_EXPERIMENTAL_ROLLOUT_REFACTOR=1，否则 train.py 报 unrecognized arguments。
5. 无 GPU 验证与文档：新增 eval_nemogym_via_api.py 提供两层扫描——golden 扫描（服务端以 run_golden=true 启动，gold patch 期望 reward 1.0，任一失败以非零码退出）与 API-policy 扫描（--policy-base-url 指向外部模型如 DeepSeek，验证 policy_base_url 覆盖的完整往返）。tests/test_nemogym_agent_function.py 提供 7 个离线单测，覆盖 /run 请求契约、响应映射、传输失败 / 超时返回 None、数据转换（不纳入仓库级 testpaths，需手动执行）。文档侧新增 docs/user-guide/nemo-gym.md、在 docs.json 注册导航、在 environments.md 表格补 NeMo-Gym 行并更新 Daytona sandbox provider 覆盖列。

关键文件：

- examples/experimental/nemo-gym/nemogym_agent_function.py（模块 环境适配器；类别 source；类型 dependency-wiring；符号 run, post_json, _resolve_session_url, build_responses_create_params）：整个集成的核心连接器：miles 每样本调用一次 run()，向 NeMo-Gym /run POST 任务并把 episode 专属 session URL 作为 policy_base_url 交给对方，实现无损策略记录；失败语义（None → reward 0.0）与采样参数映射都在这里定义。
- examples/experimental/nemo-gym/run.py（模块 训练启动器；类别 source；类型 core-logic；符号 ScriptArgs, cleanup, prepare, execute）：训练启动器：把 generate / agent function / reward / 动态采样过滤 / session server 整条 agent 链通过 train.py 扩展点接进来，并设置 MILES_EXPERIMENTAL_ROLLOUT_REFACTOR=1 门控 agentic 标志；review 中由 .sh 改写为 .py 的作品。

- examples/experimental/nemo-gym/eval_nemogym_via_api.py（模块 验证工具；类别 source；类型 entrypoint；符号 _load_instances, _run_one, _main, bounded）：不依赖 GPU 的双层验证驱动：golden 扫描（无模型，验证沙箱 + 镜像 + SWE-bench harness，reward 必须 1.0）与 API-policy 扫描（外部模型经 policy_base_url 覆盖驱动真实 episode），是保证集成质量的重要验证层。
- examples/experimental/nemo-gym/tests/test_nemogym_agent_function.py（模块 单元测试；类别 test；类型 test-coverage；符号 run_async, _capture_post, fake_post, test_run_body_carries_instance_fields_and_policy_override）：7 个离线单测，无网络、无 Docker 地锁定 /run 请求契约（实例字段顶层展开、policy_base_url、采样参数映射）、响应映射、传输失败 / 超时返回 None，以及数据转换逻辑；但因 testpaths 配置需手动执行。
- examples/experimental/nemo-gym/download_and_process_data.py（模块 数据转换；类别 source；类型 core-logic；符号 convert_to_miles_format, main）：数据处理入口：把 SWE-bench 系数据集（HF 或本地 JSONL）转为 miles prompt 数据，完整实例放入 metadata 并补 subset/split，供 NeMo-Gym 按任务选镜像与评测。
- examples/experimental/nemo-gym/nemogym_generate.py（模块 奖励函数；类别 source；类型 core-logic；符号 reward_func）：reward hook：reward 由 NeMo-Gym 在 episode 内用官方 SWE-bench harness 预计算，这里只从 sample.metadata 读取，缺失回退 0.0，是整个奖励闭环的终点。
- examples/experimental/nemo-gym/README.md（模块 示例文档；类别 docs；类型 documentation）：端到端配方文档：NeMo-Gym 服务端搭建、数据准备、启动方式、golden / API-policy 扫描验证命令、已知限制（SWE-Gym 评测缺口、Qwen3 模板软诊断、零奖励能力下限），是用户上手的主要入口。
- docs/user-guide/nemo-gym.md（模块 用户指南；类别 docs；类型 documentation；符号 POSTs）：新增用户指南页面：说明 agent-function 层集成方式、policy_base_url 上游依赖、三步上手流程与验证结论，补全 docs 用户路径。
- docs/docs.json（模块 文档导航；类别 config；类型 configuration）：文档导航配置：在 user-guide 的 environments 导航组中注册 nemo-gym 页面，保证新页面进入正式文档站点。
- docs/user-guide/environments.md（模块 环境一览；类别 docs；类型 documentation）：环境生态总表：补 NeMo-Gym 行（agent function 层）并更新 Daytona sandbox provider 覆盖列，与 PR 的生态位补全动机直接对应。

关键符号：run, post_json, _resolve_session_url, build_responses_create_params, reward_func, execute, ScriptArgs, _load_instances, _run_one, convert_to_miles_format

关键源码片段

examples/experimental/nemo-gym/run.py

训练启动器：把 generate / agent function / reward / 动态采样过滤 / session server 整条 agent 链通过 train.py 扩展点接进来，并设置 MILES_EXPERIMENTAL_ROLLOUT_REFACTOR=1 门控 agentic 标志；review 中由 .sh 改写为 .py 的作品。

```
# examples/experimental/nemo-gym/run.py
# 4 卡 GRPO 启动器（smoke 规模，2026-07-28 在 4x H200 上验证）。
# 核心作用：把整条 agent 链通过 train.py 的扩展点接进来，并设置
# MILES_EXPERIMENTAL_ROLLOUT_REFACTOR=1，否则这些 agentic 标志不会被
# 动态注册，train.py 会直接报 "unrecognized arguments"。

from dataclasses import dataclass
import os
from pathlib import Path

import miles.utils.external_utils.command_utils as U

SCRIPT_DIR = Path(__file__).resolve().parent


@dataclass
class ScriptArgs(U.ExecuteTrainConfig):
    # 已验证的 smoke 规模参数
    num_gpus_per_node: int = 4
    max_seq_len: int = 16384
    rollout_max_response_len: int = 4096
    num_rollout: int = 3
    rollout_batch_size: int = 2
    n_samples_per_prompt: int = 4
    global_batch_size: int = 8

    # NeMo-Gym 服务地址；MILES_ROUTER_EXTERNAL_HOST 用于 NeMo-Gym 宿主机
    # 无法解析 trainer 主机名时（如跨 tailnet 回连）改写 session URL
    nemo_gym_url: str = os.environ.get("NEMO_GYM_URL", "http://localhost:12000")
    router_external_host: str = os.environ.get("MILES_ROUTER_EXTERNAL_HOST", "")


def execute(args: ScriptArgs):
    # ... checkpoint / rollout / grpo / optimizer / perf 参数拼装略 ...

    agent_args = (
        # generate 层：agentic tool-call 会话生成，产出 session 记录
        "--custom-generate-function-path miles.rollout.generate_hub.agentic_tool_call.generate "
        # agent 函数层：每样本一次 POST NeMo-Gym /run，见 nemogym_agent_function.py
        "--custom-agent-function-path nemogym_agent_function.run "
        # reward 层：从 sample.metadata["reward"] 读 NeMo-Gym 的 SWE-bench 评分
        "--custom-rm-path nemogym_generate.reward_func "
        # 未触达模型的 episode 被标记 ABORTED，这里整组丢弃，避免污染训练
        "--dynamic-sampling-filter-path miles.rollout.filter_hub.dynamic_sampling_filters.check_no_
        aborted "
        "--use-session-server "
        # 绑 0.0.0.0，让 NeMo-Gym 宿主机能从任意网卡（如 tailnet 地址）回连；
        # 集群内部解析时仍映射回 localhost
        "--session-server-ip 0.0.0.0 "
```

```
        "--session-server-port 30000 "
        "--tito-model qwen3 "
    )

    extra_env_vars = {
        "PYTHONPATH": f"{args.megatron_path}:{SCRIPT_DIR}:{U.repo_base_dir}",
        # 门控 agentic 标志的动态注册；不设则 train.py 报 unrecognized arguments
        "MILES_EXPERIMENTAL_ROLLOUT_REFACTOR": "1",
        "NEMO_GYM_URL": args.nemo_gym_url,
    }
    if args.router_external_host:
        extra_env_vars["MILES_ROUTER_EXTERNAL_HOST"] = args.router_external_host

    U.execute_train(
        train_args=train_args,
        config=args,
        num_gpus_per_node=args.num_gpus_per_node,
        megatron_model_type=args.megatron_model_type,
        megatron_path=args.megatron_path,
        extra_env_vars=extra_env_vars,
    )
```

examples/experimental/nemo-gym/nemogym_generate.py

reward hook：reward 由 NeMo-Gym 在 episode 内用官方 SWE-bench harness 预计算，这里只从 sample.metadata 读取，缺失回退 0.0，是整个奖励闭环的终点。

```
# examples/experimental/nemo-gym/nemogym_generate.py
# reward hook：NeMo-Gym 环境在 episode 内部已用官方 SWE-bench harness 打分，
# 评分放在 sample.metadata["reward"]，这里只做读取，不重复计算。

from miles.utils.types import Sample


async def reward_func(args, samples: Sample | list[Sample], **kwargs) -> float | list[float]:
    """读取 NeMo-Gym 预计算的 reward，兼容单样本与样本列表两种调用形态。

    缺失时回退 0.0，对应传输失败、服务端评分出错等被 agent 函数返回
    None 的场景——此时已记录的 session 仍保留为样本，但 reward 计 0。
    """
    if isinstance(samples, list):
        return [s.metadata.get("reward", 0.0) for s in samples]
    return samples.metadata.get("reward", 0.0)
```

评论区精华

评审人为 Shi-Dong，最终 APPROVED，共两条 nit 全部由作者以 "Fixed." 解决：

1. 命名问题：Shi-Dong 指出 eval_nemogym_via_api.py 中 _build_responses_create_params 与 _post_json 并非私有函数，建议去掉下划线前缀。

作者确认修复，终版中已更名为 build_responses_create_params / post_json，作为公共导入符号被扫描脚本与测试复用。

2. 启动脚本形态：Shi-Dong 建议该示例未来大概率移出 experimental，启动脚本应改用 .py 而非 .sh。作者确认修复，最终以 run.py（typer + @U.dataclass_cli）替代原 run-qwen3-4b-instruct.sh。 两条评论均无正确性或性能争议，属于风格与工程形态建议；合并前无未解决疑虑。

- 公共函数的下划线命名 (style): nblintao 回复 "Fixed."；终版中两个函数已更名为 build_responses_create_params / post_json，并成为 eval 扫描脚本与离线测试的公共导入符号。
- 启动脚本从 .sh 改为 .py (design): nblintao 回复 "Fixed."；最终文件列表中的 run.py（typer + @U.dataclass_cli）即由原 run-qwen3-4b-instruct.sh 改写而来，为将来移出 experimental 铺路。

风险与影响

- 风险：
  1. 上游依赖未合并：整个 recipe 依赖 NVIDIA-NeMo/Gym#2166 的 policy_base_url per-request 覆盖字段；未合并期间只能运行 nblintao/Gym 的 PR 分支，上游字段改名或行为变动（如 ng-rollout 前缀逻辑变化）会直接破坏 nemogym_agent_function.py 的请求契约，需持续跟踪合并时间线。
  2. 实验开关门控：MILES_EXPERIMENTAL_ROLLOUT_REFACTOR=1 未设置时 train.py 会以 "unrecognized arguments" 失败；run.py 已设置，但用户改用自己的启动脚本时容易遗漏，报错信息不够直观。
  3. SWE-Gym 评分缺口：官方 swebench 包缺少多个 SWE-Gym repo 的 eval spec（make_test_spec 抛 KeyError: 'getmoto/moto'），SWE-Gym episode 能跑但评分阶段报错记 0 分；用户直接训练 SWE-Gym 数据集会得到系统性零奖励且原因隐晦。
  4. 零奖励训练信号：4B 策略在 SWE-bench 上 solve 率为 0，GRPO advantage 恒为 0（rollout/zero_std 触发），可能被误判为管线缺陷；README 已标注为能力下限而非缺陷，但无自动防护。
  5. 软诊断指标失真：Qwen3 模板重渲染 assistant 历史时插入空 ，导致 rollout/tito_session_mismatch_rate 恒为 1.0；训练 token 与 loss mask 来自引擎记录的 token id 不受影响，但若该指标被用于质量过滤会产生误判。
  6. 测试未纳入 CI：仓库级 pytest 的 testpaths=./tests 不收集 examples/experimental/nemo-gym/tests，7 个离线单测需手动执行，缺少回归保护。
  7. 网络暴露面：session server 绑 0.0.0.0:30000 面向外部 NeMo-Gym 主机开放；MILES_ROUTER_EXTERNAL_HOST 配置错误会导致策略请求超时，且集群内其他主机可访问该端口。 - 影响：影响范围：纯增量变更——不修改任何核心库文件，只新增 examples/experimental/nemo-gym 与 docs 下的内容，对既有训练管线零影响；属于中低风险。对用户：获得 NVIDIA NeMo-Gym 生态（SWE-bench 代理训练）的一键式 recipe，含数据转换、启动、无 GPU 验证全套工具链；与 Harbor、OpenEnv 并列成为第三个 agent function 层外部环境接入范本。对系统：session server 首次承接来自集群外的策略流量（NeMo-Gym 宿主机回连），网络与安全面略有扩展；无损 token 记录

能力成为第三方环境集成的通用底座。对团队：确立了 " 上游 PR 优先、不 fork、不 submodule" 的第三方环境集成策略，取代了 #1918 的 fork 方案；文档表新增 NeMo-Gym 行并更新 sandbox provider 覆盖说明。影响程度：中等——实验示例本身影响有限，但其确立的接入范式与上游 PR 依赖可能影响后续环境类集成决策。 - 风险标记：上游 PR 未合并，实验开关门控，SWE-Gym 评分缺口，零奖励误导风险，测试未纳入 CI

关联脉络

- PR #1918 （标题未知，PR body 提及）移除 fork-based generate-function 集成：PR body 明确说明本 PR 取代 #1918 移除的 fork 方案：旧方案对消息文本重新分词并依赖两个个人 fork submodule，新方案走上游 NeMo-Gym + session server 无损记录，是同一功能线的演进。
- PR #1790 openenv/tbench2: score the shared-server leg natively; retire the adapter compensation: 同为 agent function 层的外部环境集成（OpenEnv），其 openenv_agent_function.py 与 scan_golden.py 与本 PR 的 agent 函数、eval 扫描结构同型，是环境接入的第三条范式，可对照阅读。
- PR #1759 (2/2) refactor(session): assemble training samples on the session server; records never leave it: session server 在服务端组装训练样本、records 不离开服务器，是本 PR 中 " 每个 episode 一个 session URL、token id + logprobs + loss mask 无损记录 " 落地的基础设施前提。