

PR #1739 完整报告

radixark/miles

feat: add Verifiers rollout integration

合并时间: 2026-08-07 06:44

原文链接: <http://prhub.com.cn/radixark/miles/pull/1739>

执行摘要

- 一句话: 新增 Verifiers 环境训练适配器, 示例化接入分组奖励 rollout
- 推荐动作: 值得精读。重点看四点: 1) 所有权边界设计——Verifiers/Renderers/Miles 三方职责划分, 以及 "不支持即 400 显式失败" 的 fail-fast 模式; 2) `_train_client` 中本地 tokenizer 与 Renderers 注册身份的桥接技巧; 3) "把集成收口到 `examples/`、核心零改动" 的提交历史, 以及 `eval-interval workaround` 的取舍过程; 4) 依赖版本上限 (Verifiers<0.2.1) 的处理方式——错误信息直接指向 OpenAI pin 冲突, 便于后续解锁。该 PR 对后续新增环境连接器有直接参考价值。

功能与动机

PR body 说明目标是 "Add an opt-in Verifiers V1 rollout adapter selected by one new argument `--verifiers-config`", 让用户能在 Prime Intellect Verifiers 的 taskset 环境 (如 `code-golf-v1`、`reverse-text-v1`) 上直接训练, "Verifiers runs the environment lifecycle, including runtimes, user simulators, tools, multi-turn episodes, retries, per-rollout rewards, and group rewards"。设计上刻意 "follows Miles' existing rollout-function extension point instead of changing shared sample conversion or advantage code", 避免侵入共享 sample 转换与优势计算; 依赖上限是有意的: "Verifiers 0.2.1 requires OpenAI >=2.9 while the current SGLang 0.5.15 dependency pins OpenAI 2.6.1"。不支持的能力选择显式失败: "Unsupported combinations fail explicitly rather than retrying indefinitely", 尤其多分支 trace (含 compaction) 会被拒绝, 因为 Miles 展平多训练样本时不保留 Verifiers trace 分组边界, 接受这类 trace 会静默算出错误的组相对优势。

实现拆解

1. 接入入口与参数接线: 新增启动脚本 `examples/experimental/verifiers/run.py`, 通过通用插槽 `--rollout-function-path verifiers_rollout.VerifiersRolloutFn` (无 `MILES_EXPERIMENTAL_ROLLOUT_REFACTOR=1` 时用 `.generate_rollout`) 选择适配器, 并用 `--disable-rollout-global-dataset` 关闭 Miles 自身 prompt 数据集加载; Verifiers 的 `EnvConfig TOML` 路径经环境变量 `VERIFIERS_CONFIG` 注入 rollout actor (`execute_train(extra_env_vars=...)`), 避免新增 Miles 核心参数。
2. 适配器主体 `examples/experimental/verifiers/verifiers_rollout.py` (887 行):
 - `_load_config_data` 强制 `.toml` 后缀与 Python 3.11; `_check_version` 做可选依赖版本门禁 (`verifiers>=0.2.0,<0.2.1`、`renderers>=0.1.8`), 上限错误信息直接指向 OpenAI pin 冲

突；_validate_args 在构造阶段拒绝 partial rollout、自定义 chat template、多模态、OPD、路由重放、indexer 重放等组合；_renderer_identity 从本地 checkpoint 路径反推 Renderers 注册身份；_train_client 返回 TrainClient 子类，将本地 tokenizer 文件与规范模型身份解耦，并把 streaming/aux 路由等不支持请求统一转成 400 ProviderError。

3. 传输层 MilesSGLangTransport: 把 Renderers 的 /inference/v1/generate wire 格式翻译成 Miles SGLang /generate 调用；_sampling_params 做字段名映射（max_tokens/min_tokens/seed → max_new_tokens/min_new_tokens/sampling_seed）、合并 stop token、用 rollout_max_context_len/rollout_max_response_len 做双重截断，超长时抛 OverlongPromptError；session 缓存用上限 10,000 的 OrderedDict 做 LRU 淘汰，避免内存无界增长。
4. trace→Sample 转换: trace_to_samples/trace_to_sample 把 Verifiers trace (branches、rewards、metrics、task) 转成普通 Miles Sample 组，保留 tokens、rollout_log_probs、loss_mask、group_index、routing_key，并把 verifiers 的 rewards/metrics 写入 metadata["verifiers"]；reward_key 场景下奖励保持 dict 结构，统一用 sample.get_reward_value(args) 取值；_validate_group_reward_sample_counts 与 _trace_eval_reward 覆盖评估侧组奖励。
5. 启动脚本的 eval workaround: Miles 断言设置 --eval-interval 时必须配置 eval 数据集，而本集成的评估由适配器接管，因此 run.py 传入占位 --eval-prompt-data verifiers-taskset {config_path}，内置加载器永不打开该路径；代码注释与文档均标注这是值得在 Miles 侧修复的临时方案。
6. 测试与 CI 配套: tests/fast/rollout/test_verifiers_rollout.py (698 行) 覆盖参数接线、配置加载、版本上限、trace 转换、结构化奖励、传输层边界与不支持组合的 fail-fast；test_verifiers_runtime.py 用真实发布版 verifiers 0.2.0/renderers 0.1.8 验证 V1 环境契约与 TrainClient 端到端请求；tests/e2e/long/test_qwen3_0.6B_verifiers.py 注册 900s 双卡 H200 长测，prepare 阶段安装 requirements、prime CLI、clone verifiers v0.2.0 并 prime env install code-golf-v1，verify 阶段校验 12 样本的 group_index、奖励有限性、log_probs/loss_mask 对齐、fastest 组奖励和恰为 0.5；Python <3.11 或缺少可选依赖时测试以 importorskip/ 模块级 skip 优雅跳过。
7. 文档配套: docs/user-guide/verifiers.md、示例 README、docs/docs.json 在 Environments 分区注册新页面、docs/user-guide/environments.md 连接器总览表新增 Verifiers 行（注明接入方式为 rollout function）。

关键文件：

- examples/experimental/verifiers/verifiers_rollout.py (模块 适配器；类别 source；类型 core-logic；符号 VerifiersRolloutFn, generate_rollout, trace_to_samples, trace_to_sample) : 887 行适配器主体: EnvConfig 加载、依赖版本门禁、Renderers 身份推断、TrainClient 子类、MilesSGLangTransport 传输层、trace→Sample 转换与不支持组合校验，是本 PR 的核心交付物。
- examples/experimental/verifiers/run.py (模块 启动脚本；类别 source；类型 entrypoint；符号 ScriptArgs, prepare, execute, main) : 启动入口：演示 plug-point 接线、VERIFIERS_CONFIG 注入、自动生成 EnvConfig TOML，以及 eval-interval 占位 workaround；是 " 零核心改动 " 形态的载体。

- tests/fast/rollout/test_verifiers_rollout.py (模块 单元测试; 类别 test; 类型 test-coverage; 符号 _args, _branch, _trace, _wiring_args) : 698 行核心单元测试: 接线校验、配置加载、版本上限、trace→sample 转换、结构化奖励、传输层边界、不支持组合 fail-fast; 是适配器行为契约的硬锁定。
- tests/fast/rollout/test_verifiers_runtime.py (模块 运行时测试; 类别 test; 类型 test-coverage; 符号 _args, test_minimal_env_config_uses_the_v1_environment_contract, _Rendered, message_token_spans) : 用真实发布版 verifiers 0.2.0/renderers 0.1.8 验证 V1 环境契约 (EnvConfig/Environment) 与 TrainClient 经 MilesSGLangTransport 的端到端请求路径, 证明适配器对接的是上游公开 API 而非假桩。
- tests/e2e/long/test_qwen3_0.6B_verifiers.py (模块 端到端测试; 类别 test; 类型 test-coverage; 符号 prepare, execute, verify) : 双卡 H200 长测 (900s) : 完整跑通 Qwen3-0.6B 对 code-golf-v1 的真实 optimizer 更新, 并校验 12 个样本的组结构、奖励有限性、log_probs/loss_mask 对齐与 fastest 组奖励和; 是集成质量的硬证据。
- docs/user-guide/verifiers.md (模块 用户文档; 类别 docs; 类型 documentation) : 用户指南: 安装 (requirements + prime CLI + prime env install)、配置 (EnvConfig TOML)、所有权边界、与 Miles 参数对照表、限制说明; 并明确 0.2.1 的 OpenAI pin 冲突。
- examples/experimental/verifiers/README.md (模块 示例文档; 类别 docs; 类型 documentation) : 示例 README: 说明适配器作为 rollout function 的工作原理、安装、配置、运行方式与不支持项, 是用户从示例入手的第一入口。
- docs/docs.json (模块 文档配置; 类别 config; 类型 configuration) : 在 docs 导航的 Environments 分区注册 user-guide/verifiers 页面, 是文档导航的纯配置变更 (+2/-1)。
- examples/experimental/verifiers/requirements.txt (模块 依赖清单; 类别 config; 类型 dependency-wiring) : 固定可选依赖范围: verifiers>=0.2.0,<0.2.1、renderers>=0.1.8、openai-agents<0.5; 注释明确说明 0.2.1 与 openai 2.6.1 pin 的冲突原因。
- docs/user-guide/environments.md (模块 用户文档; 类别 docs; 类型 documentation) : 连接器总览表新增 Verifiers (Prime Intellect) 行, 注明接入方式为 rollout function, 与 Harbor/OpenEnv/NeMo-Gym/Strands/tau-bench 并列。

关键符号: generate_rollout, VerifiersRolloutFn, trace_to_samples, trace_to_sample, MilesSGLangTransport, _sampling_params, _train_client, _renderer_identity, _import_verifiers, _check_version, _validate_args, _config_path, _load_config_data, _finish_reason, _sglang_worker_urls

关键源码片段

examples/experimental/verifiers/verifiers_rollout.py

887 行适配器主体: EnvConfig 加载、依赖版本门禁、Renderers 身份推断、TrainClient 子类、MilesSGLangTransport 传输层、trace→Sample 转换与不支持组合校验, 是本 PR 的核心交付物。

```
# 关键实现: 自定义 TrainClient 子类, 把「本地 tokenizer 文件」与「Renderers 注册表
# 中的规范模型身份」解耦。Miles 的 checkpoint 通常是本地快照, 路径本身不带注册身份。
def _train_client(runtime, args: Namespace, model: str, pool_size: int, *, router_args=None):
    # tokenizer_source 优先取 --sglang-tokenizer-path, 再退回模型名; identity 用于
```

```

# 从本地路径反推 Renderers 能识别的模型 id。
tokenizer_source = getattr(args, "sglang_tokenizer_path", None) or model
identity = _renderer_identity(model) or _renderer_identity(tokenizer_source)

class TrainClient(runtime.TrainClient):
    @staticmethod
    def _unsupported_request(kind: str):
        # 不支持的请求统一转成 400 ProviderError, 让上层显式失败而不是静默重试。
        return runtime.ProviderError(f"{_UNSUPPORTED_ERROR_PREFIX} {kind}.", status_code=400)

    async def get_response(self, *args, **kwargs):
        try:
            return await super().get_response(*args, **kwargs)
        except NotImplementedError as error:
            # 非 Chat 方言等能力缺口: 直接转 400, 避免无限重试。
            raise runtime.ProviderError(
                f"{_UNSUPPORTED_ERROR_PREFIX} this request: {error}", status_code=400)
            from error
        except ValueError as error:
            if "does not support tools" not in str(error):
                raise
            # 工具调用需要特定模型的 renderer; 提示用户改用注册过的模型身份。
            raise runtime.ProviderError(
                f"{_UNSUPPORTED_ERROR_PREFIX} tools with this renderer: {error} "
                "Use a Renderers-registered model identity in --hf-checkpoint or "
                "--sglang-tokenizer-path.", status_code=400) from error

    async def relay(self, *args, **kwargs):
        raise self._unsupported_request("streaming requests")

    async def relay_aux(self, *args, **kwargs):
        raise self._unsupported_request("auxiliary dialect routes")

    def _renderer_pool(self, requested_model, *, chat_template_kwargs=None):
        # identity 未知时直接走 Renderers 默认路径; 否则用本地 tokenizer 加载,
        # 再改写 name_or_path 为规范身份, 让 Renderers 找到正确的 renderer。
        if identity is None:
            return super()._renderer_pool(requested_model, chat_template_kwargs=chat_template_kwargs)
        if self._pool is None:
            from renderers import RendererPool, create_renderer
            from renderers.base import load_tokenizer

            source = self.renderer_model_name or requested_model

            def factory():
                tokenizer = load_tokenizer(source)
                tokenizer.name_or_path = identity

```

```

        return create_renderer(tokenizer, self.config, chat_template_kwargs=chat_
            template_kwargs)

        self._pool = RendererPool(factory, size=self.pool_size)
        return self._pool

    return TrainClient(
        MilesSGLangTransport(args, router_args=router_args),
        pool_size=pool_size,
        renderer_model_name=tokenizer_source,
    )

```

examples/experimental/verifiers/run.py

启动入口：演示 plug-point 接线、VERIFIERS_CONFIG 注入、自动生成 EnvConfig TOML，以及 eval-interval 占位 workaround；是 " 零核心改动 " 形态的载体。

```

def execute(args: ScriptArgs):
    # 默认配置只写 taskset id，其余 (harness、runtime、judges、retries) 交给 Verifiers 默认值。
    config_path = Path(args.verifiers_config)
    if not config_path.exists():
        config_path.write_text(f'[taskset]\nid = "{args.taskset_id}"\n')

    ckpt_args = (
        f"--hf-checkpoint {args.hf_checkpoint} "
        f"--sglang-tokenizer-path {args.sglang_tokenizer_path} "
        f"--ref-load {args.ref_load} "
        f"--save {args.save_dir} "
        "--save-interval 1000 "
    )

    # Verifiers 拥有 taskset，因此 Miles 不加载 prompt 数据；适配器通过通用
    # plug-point --rollout-function-path 接入，以裸模块名解析 (PYTHONPATH 携带本目录)。
    rollout_fn = (
        "verifiers_rollout.VerifiersRolloutFn"
        if os.environ.get("MILES_EXPERIMENTAL_ROLLOUT_REFACTOR") == "1"
        else "verifiers_rollout.generate_rollout"
    )

    rollout_args = (
        f"--rollout-function-path {rollout_fn} "
        "--disable-rollout-global-dataset "
        f"--num-rollout {args.num_rollout} "
        f"--rollout-batch-size {args.rollout_batch_size} "
        f"--n-samples-per-prompt {args.n_samples_per_prompt} "
        f"--over-sampling-batch-size {args.rollout_batch_size} "
        f"--rollout-max-response-len {args.rollout_max_response_len} "
        f"--rollout-max-context-len {args.rollout_max_context_len} "
        "--rollout-temperature 0.8 "
        f"--rollout-num-gpus-per-engine 1 "
    )

```

```
# Workaround: Miles 断言设置 --eval-interval 时必须配置 eval 数据集，而本集成
# 的评估由适配器接管；于是用占位符命名 taskset 并指向其 EnvConfig，内置加载器
# 永远不会打开该路径。这是明确标注过的临时方案，值得以后在 Miles 侧修复。
```

```
eval_args = (
    (
        f"--eval-interval {args.eval_interval} "
        f"--n-samples-per-eval-prompt {args.n_samples_per_eval_prompt} "
        f"--eval-prompt-data verifiers-taskset {config_path} "
    )
    if args.eval_interval
    else ""
)
```

```
# grpo/optimizer/perf/sglang/misc 参数组为常规 GRPO 冒烟配置，从略
```

```
U.execute_train(
    train_args=(
        f"{ckpt_args}{rollout_args}{eval_args}{grpo_args}{optimizer_args}{perf_args}{sglang_
args}{misc_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={
        # 配置路径通过环境变量注入 rollout actor，适配器从 VERIFIERS_CONFIG
        # 读取，而不是新增一个 Miles 核心参数。
        "PYTHONPATH": f"{args.megatron_path}:{SCRIPT_DIR}:{U.repo_base_dir}",
        "VERIFIERS_CONFIG": str(config_path),
    },
)
```

评论区精华

1. 核心参数校验改动是否应纳入本 PR (Shi-Dong vs nblintao, 影响最大) : Shi-Dong 对放宽 --eval-interval 的 eval 数据集断言表示 "I am not entirely sure about this change... I think it's debatable and merits a standalone PR", 建议本 PR 用 dummy eval_datasets 绕开; nblintao 同意 ("Yeah, I think we can use dummy eval_datasets to workaround for now")。最终 commit "Move the Verifiers integration into examples/" 更进一步把整个适配器移出 miles/, 实现 "this PR touches no core file at all".
2. 配置冲突必须报错而非静默覆盖 (nblintao) : "Raise exception if user already set args.rollout_function_path or args.rollout_global_dataset (e.g. multi-lora)". 最终以 _validate_args 在构造阶段拒绝冲突组合, 并由 test_unsupported_wiring_fails_before_a ny_episode_runs 参数化锁定。
3. 机器人评审提出两个真问题并已修复: eval 平均奖励直接访问 sample.reward, 当 reward 为 dict 时 TypeError (改为 sample.get_reward_value(self.eval_args), 补 test_eval_extracts_structured_miles_reward); _seen_sessions 集合无界增长 (改为

10,000 上限的 OrderedDict LRU 淘汰, 补 test_transport_bounds_seen_sessions) 。

4. 文档命名与分区 (nblintao) : "'Verifiers' can mean general verifiers in RL setup", 要求改为 Verifiers (Prime Intellect); cli-reference 段落改名 Environment Integrations; 并按要求 rebase 到 #1833 的 Environments 文档分区。
- 核心参数校验改动是否应纳入本 PR / 集成是否应收口到 examples/ (design): 采用 dummy eval-prompt-data 占位符 workaround; 集成整体迁移到 examples/experimental/verifiers , miles/ 核心目录不改。
- eval 平均奖励直接相加结构化 reward 可能 TypeError (correctness): head 已改用 get_reward_value 提取奖励后再求平均, 并新增 test_eval_extracts_structured_miles_reward 覆盖 dict 奖励。
- MilesSGLangTransport._seen_sessions 无界增长 (performance): head 改为 OrderedDict 缓存, 上限 10,000, 命中时 move_to_end 做 LRU 刷新, 满时淘汰最旧项; 新增 test_transport_bounds_seen_sessions。
- --verifiers-config 与自定义 rollout-function-path / multi-LoRA 冲突须报错 (design): 以 _validate_args 在构造阶段拒绝冲突组合, 并由 test_unsupported_wiring_fails_before_any_episode_runs 参数化锁定全部不支持项。
- 文档标题 'Verifiers' 与通用 verifiers 概念混淆 (documentation): 文档页改为 'Verifiers (Prime Intellect)' 并放入用户指南的 Environments 分区。
- cli-reference 段落改名为 Environment Integrations (documentation): 已按要求改名; 集成最终移入 examples/ 后, 相关核心参数段落随 arguments.py 变更一并移除。
- rebase 到 #1833 的 Environments 文档分区 (documentation): 最终 docs.json 在 Environments 分区注册 user-guide/verifiers, environments.md 连接器表新增 Verifiers 行。

风险与影响

- 风险:
 1. 可选依赖与版本上限: requirements.txt 固定 verifiers $\geq$ 0.2.0,<0.2.1、renderers $\geq$ 0.1.8、openai-agents<0.5, 上限源于 Verifiers 0.2.1 要求 openai $\geq$ 2.9 而 SGLang 0.5.15 固定 openai==2.6.1; _check_version 在超限时直接抛错并注明原因, SGLang 升级 openai pin 后需同步放开。手动接线时若缺少依赖, _import_verifiers 会以 _optional_dependency_error 失败, 报错指向 pip install -e '[verifiers]'。
 2. Python 3.11 门槛: _load_config_data/_import_verifiers 在 <3.11 直接抛错, 快速测试用模块级 skip 跳过; 3.10 环境已验证 import 路径干净, 但文档需明确提示版本要求。
 3. E2E 长测依赖外部资源: test_qwen3_0.6B_verifiers.py 在 prepare 阶段 git clone verifiers v0.2.0、uv tool install prime==0.6.19、prime env install code-golf-v1, 外部 CLI 或网络变化会直接导致 CI 失败; 测试时长 900s (2xH200), 维护成本较高。
 4. eval workaround 脆弱: run.py 用占位 --eval-prompt-data verifiers-taskset {config_path} 绕过 Miles 的 eval 数据集断言, 依赖 " 内置加载器永不打开该路径 " 的隐式约定; 后续 Miles 侧解析逻辑或断言变化可能打破, 社区已标注值得在 Miles 侧修

复。

5. 能力边界为已知形态：partial rollout、自定义 chat template、多模态、OPD、路由 / 索引器重放、多分支 trace (含 compaction) 在构造阶段显式失败 (ValueError/400 ProviderError)，不会静默产出错误样本；但启用 compaction 的 taskset 当前无法使用，且工具调用依赖模型在 Renderers 注册表中存在对应 renderer，本地私有微调模型需要传注册过的 --sglang-tokenizer-path。 - 影响：用户侧：获得在 Prime Intellect Verifiers 环境 (code-golf-v1、reverse-text-v1 等) 上训练的能力，支持工具调用、多轮、用户模拟器与组奖励；要求 Python 3.11+ 与 prime CLI，环境通过 prime env install 安装；Miles 既有语义 (采样、LoRA、动态过滤、路由、resume 位置) 全部保留。系统侧：最终合入版本对 miles/ 核心零改动，全部代码位于 examples/experimental/verifiers/，不影响既有 rollout、sample 转换与 advantage 路径，属低侵入的新功能。团队与规范侧：确立了 " 环境集成 = examples/ + 通用 plug-point 参数 " 的仓库形态，与 Harbor/OpenEnv/NeMo-Gym/Strands/tau-bench 一致；docs Environments 分区新增 Verifiers 入口。CI 侧：新增 fast 测试 (<3.11 优雅跳过) 与 long 双卡 E2E (900s)，PR 仅挂 run-ci-short 标签，长测并非每次合入都全量跑，需留意持续维护成本。 - 风险标记：依赖版本上限约束 Verifiers<0.2.1, Python 3.11 专用模块，E2E 依赖外部 git clone 与 prime CLI, eval 数据集断言 workaround 脆弱，多分支 trace 显式拒绝 (能力边界)

关联脉络

- PR #2233 examples: rename swe-agent to swe-agent-harbor-docker: 同属 examples/ 环境集成收口脉络，均改动 docs/user-guide/environments.md 与 examples 结构，确立 " 环境集成 = examples/ + 通用 plug-point 参数 " 的仓库形态。
- PR #1913 E2B sandbox backend (E2B Cloud / self-hosted AgentENV) + dedicated AgentENV recipe: examples/experimental/openenv 沙箱后端接入的先例，与本 PR 一样通过 examples/ 扩展点接入新环境族；environments.md 明确 " 沙箱提供方 " 与 " 环境连接器 " 是两个不同维度。
- PR #1919 examples: add swe-agent-harbor-daytona (Harbor sandboxes on Daytona): examples/experimental 环境接入的另一个先例，验证了本 PR 采用的 examples 目录形态、README 与文档接入模式。
- PR #1833 (标题未提供) docs: Environments 文档分区：nblintao 在 review 中要求本 PR rebase 到该 PR，将 Verifiers 文档放入其新增的 Environments 分区；最终 docs.json 结构即为该分区形态。
- PR #1487 (标题未提供) rollout-function 集成先例：PR body 明确说明 "The implementation follows the rollout-function integration pattern used by prior merged work such as #1487"。
- PR #1633 (标题未提供) rollout-function 集成先例：同 #1487，PR body 提及的另一个 rollout-function 扩展模式先例。