

PR #6302 完整报告

verl-project/verl

[BREAKING][tool, data] refactor: remove curriculum sampler + dynamic dataset + tool examples

合并时间: 2026-05-09 21:54

原文链接: <http://prhub.com.cn/verl-project/verl/pull/6302>

执行摘要

- 一句话: 移除废弃工具、课程采样器和动态数据集
- 推荐动作: 该 PR 是值得参考的代码清理范例: 它大规模移除了废弃代码并扁平化目录, 附带全面的引用更新和测试清理。建议内部开发者检查自己的分支是否依赖了被删除的组件, 并及时调整。对于仓库维护者, 这次重构为进一步解耦和标准化工具注册机制铺平了道路。

功能与动机

根据 PR body, 这些组件是 'dormant or duplicated plugin entry points that no in-tree code uses', 移除它们以简化代码库、减少维护成本、消除混淆。

实现拆解

1. 移除参考工具实现: 删除了 5 个工具文件 (gsm8k_tool.py、geo3k_tool.py、image_zoom_in_tool.py、sandbox_fusion_tools.py、search_tool.py) 及辅助模块 search_r1_like_utils.py。这些工具均继承 BaseTool, 实现了 OpenAI 工具模式、异步执行、奖励计算等, 但不再被任何内部代码引用。
2. 移除实验性数据集组件: 删除了 sampler.py (AbstractSampler、AbstractCurriculumSampler) 和 dynamicgen_dataset.py (DynamicGenDataset、AbstractDataGenerator), 以及配置块 data.sampler 和 data.datagen。这些代码原本用于课程学习与动态数据生成, 但未被集成到训练主流程。
3. 目录扁平化: 将 verl/tools/utils 下的 function_tool.py、tool_registry.py 等移到 verl/tools 根目录, 更新所有导入路径, 消除多余目录层级。
4. 简化 create_rl_sampler: 移除了插件加载逻辑, 现在只支持随机采样和顺序采样两个分支, 直接内联判断。
5. 清理引用: 修改了 main_ppo.py 和多个配置文件, 移除对已删除模块的导入和训练钩子 (如 curriculum sampler 的 update 调用); 删除了对应的测试文件和 init.py 文件, 更新文档。

关键文件:

- verl/tools/image_zoom_in_tool.py (模块 工具; 类别 source; 类型 deletion; 符号 PoolMode, TokenBucketWorker, init, acquire): 移除了一个图像放大工具 (ImageZoomInTool), 该工具包含 Ray 令牌桶限流器和视觉执行池, 不再被使用。

- verl/tools/search_tool.py (模块 工具; 类别 source; 类型 deletion; 符号 PoolMode, TokenBucketWorker, init, acquire) : 移除了搜索工具 (SearchTool) , 包含搜索执行池和关联的搜索 API 调用函数。
- verl/tools/sandbox_fusion_tools.py (模块 工具; 类别 source; 类型 deletion; 符号 PoolMode, TokenBucketWorker, init, acquire) : 移除了沙盒融合工具 (SandboxFusionTool) , 用于在沙盒中执行代码并评分。
- verl/tools/gsm8k_tool.py (模块 工具; 类别 source; 类型 deletion; 符号 Gsm8kTool, init, get_openai_tool_schema, create) : 移除了一个参考工具 Gsm8kTool, 用于计算 GSM8K 题目奖励, 不再需要。
- verl/experimental/dynamic_dataset/dynamicgen_dataset.py (模块 数据集; 类别 source ; 类型 deletion; 符号 AbstractDataGenerator, init, generate, MockDataGenerator) : 移除了动态数据集模块 (DynamicGenDataset 和 AbstractDataGenerator) , 这些代码用于在训练间动态生成数据, 但从未集成到主流程。
- verl/experimental/dataset/sampler.py (模块 采样器; 类别 source; 类型 deletion; 符号 AbstractSampler, init, AbstractCurriculumSampler, update) : 移除了实验性课程采样器 (AbstractSampler、AbstractCurriculumSampler) , 这些采样器原本用于课程学习。
- verl/trainer/main_ppo.py (模块 训练器; 类别 source; 类型 dependency-wiring) : 移除了对 curriculum sampler 的导入和训练循环中的 update 钩子调用, 同时更新了工具模块的导入路径。
- tests/utils/dataset/test_create_rl_sampler_on_cpu.py (模块 测试; 类别 test; 类型 deletion; 符号 RandomCurriculumSampler, init, iter, len) : 移除了对应的测试文件, 因为被测试的随机课程采样器已删除。

关键符号: create_rl_sampler, update

关键源码片段

verl/tools/image_zoom_in_tool.py

移除了一个图像放大工具 (ImageZoomInTool) , 该工具包含 Ray 令牌桶限流器和视觉执行池, 不再被使用。

```
# 以下是被移除的 TokenBucketWorker 类。
# 它作为一个基于 Ray 的令牌桶限流器, 被多个工具执行池重复实现。
#
由于所有相关工具 (ImageZoomInTool、SearchTool、SandboxFusionTool) 均已删除, 该类也不再需要。
```

```
@ray.remote(concurrency_groups={"acquire": 1, "release": 10})
class TokenBucketWorker:
    """Ray actor for rate limiting using token bucket algorithm."""

    def __init__(self, rate_limit: int):
        self.rate_limit = rate_limit
        self.current_count = 0 # 用于观测
        self._semaphore = threading.Semaphore(rate_limit)
```

```

@ray.method(concurrency_group="acquire")
def acquire(self):
    """获取一个令牌。"""
    self._semaphore.acquire()
    self.current_count += 1

@ray.method(concurrency_group="release")
def release(self):
    """释放一个令牌。"""
    self._semaphore.release()
    self.current_count -= 1

def get_current_count(self):
    """获取当前已获取的令牌数。"""
    return self.current_count

```

verl/experimental/dynamic_dataset/dynamicgen_dataset.py

移除了动态数据集模块（DynamicGenDataset 和 AbstractDataGenerator），这些代码用于在训练间动态生成数据，但从未集成到主流程。

- # 以下是被移除的 AbstractDataGenerator 基类和 MockDataGenerator 实现。
- # 它们用于在训练迭代之间动态生成或扩充数据集。
- # 由于 DynamicGenDataset 未被任何训练器集成，整个框架被删除。

```

from abc import ABC, abstractmethod
import datasets

class AbstractDataGenerator(ABC):
    """抽象数据生成器，子类必须实现 generate 方法。"""

    def __init__(self, config: DictConfig):
        self.config = config

    @abstractmethod
    def generate(self, dataset: Dataset) -> datasets.Dataset:
        """生成或处理数据集。子类需实现此方法。"""
        pass

class MockDataGenerator(AbstractDataGenerator):
    """占位数据生成器，仅追加第一个数据点，用于测试。"""

    def __init__(self, config: DictConfig = None):
        super().__init__(config)

    def generate(self, dataset: Dataset) -> datasets.Dataset:
        print("MockDataGenerator: No operation performed on the dataset.")
        return dataset.dataframe.select([0])

```

评论区精华

审核者 `gemini-code-assist[bot]` 审查后表示 'I have no feedback to provide.'，未提供具体意见。仓库维护者 `wuxibin89` 直接批准了 PR，未引发讨论。由于 PR 目标是清理废弃代码，且已通过 CI，未产生争论点。

- 代码审查 (other): 无争议，直接合并。

风险与影响

- 风险：主要风险来自 BREAKING 变更：如果外部项目或未跟踪脚本依赖了被删除的工具或数据集组件，升级后会出现导入错误。另外，目录扁平化 (`verl/tools/utils` → `verl/tools`) 可能破坏一些隐式导入。但 PR 声称所有删除的代码在内部已无引用，且 CI 通过，因此对主流训练流程影响较小。潜在风险点包括：
 - 如果用户自定义了依赖 curriculum sampler 的训练循环（调用 `dataset.update`），则需要移除该调用。
 - 被删除的工具曾是参考实现，可能被下游示例仓库或论文复现代码引用。
 - 影响：对内部系统：代码量减少约 1700 行，降低了维护成本，工具模块更加清晰，只有一个目录层。对开发者：`create_rl_sampler` 更简单，移除了复杂的插件加载；配置项 `data.sampler` 和 `data.datagen` 不再有效。对外部用户：如果使用了被删除的工具类（如 `SearchTool`、`SandboxFusionTool`）或实验性数据集（`DynamicGenDataset`），需要在升级前迁移或者自行维护。本次变更还可能影响基于 `verl` 的论文实验复现。
- 风险标记：BREAKING 变更，大量组件移除，外部使用者影响，目录结构扁平化

关联脉络

- PR #6300 [tool] refactor: tools will be initialized in AgentLoopWorker: 本 PR 是工具模块清理的后续，6300 已重构工具初始化并移除 MCP 支持，本 PR 进一步移除了不再需要的旧工具实现。
- PR #6189 [tool] feat: simpler function-based tool registration: 6189 引入了 `@function_tool` 装饰器简化工具注册，本 PR 移除了被替代的手写工具实现。