

PR #1941 完整报告

THUDM/slime

Add multi-sample test

合并时间: 2026-05-25 16:00

原文链接: <http://prhub.com.cn/THUDM/slime/pull/1941>

执行摘要

- 一句话: 新增 fanout 端到端测试, 验证单 prompt 变长多样本训练链路
- 推荐动作: 值得阅读以理解 slime 中 'compact rollout' 模式的设计约束和集成方式。特别是 rollout_id 共享对 step splitter、reducer、loss aggregation 的影响。

功能与动机

此前 CPU 单元测试已覆盖 rollout-aware step splitter、per-rollout mean reducer 等独立环节, 但始终缺少一个端到端的训练验证。这是首次在真实训练流程中检验 custom_generate 返回 list[Sample] 并共享 rollout_id 的完整链路, 确保框架不会因为样本数不均匀而 silent 地退化到错误的分组归一化。

实现拆解

1. 新增 slime/rollout/_fanout_test_helpers.py, 包含 compact_generate (异步生成函数, 对每个 prompt 调用一次 sglang 生成 base_sample, 再 deepcopy 出 $N = 1 + (\text{index} \% 3)$ 个 siblings, 全部设置相同 rollout_id) 和 grpo_normalize_by_group_index (替代默认的 _post_process_rewards, 按 sample.group_index 分组进行 GRPO 标准化)。
2. 新增 tests/test_qwen2.5_0.5B_fanout_short.py, 定义 prepare (下载模型和数据集) 和 execute (运行训练命令, 使用 --custom-generate-function-path 和 --custom-reward-post-process-path 注入辅助函数)。测试通过环境变量计数器验证自定义 generate 被调用了 num_rollout * rollout_batch_size 次, 没有静默回退。
3. 修改 .github/workflows/pr-test.yml 和 .github/workflows/pr-test.yml.j2, 在 e2e-test-sglang 矩阵中加入该测试。
4. 修改 pyproject.toml, 在 [tool.ruff] 下添加空的 [tool.ruff.lint] section, 以适配较新版本的 ruff 配置格式。

关键文件:

- slime/rollout/_fanout_test_helpers.py (模块 rollout 辅助; 类别 source; 类型 dependency-wiring; 符号 compact_generate, grpo_normalize_by_group_index): 核心辅助模块, 提供 compact_generate 和 grpo_normalize_by_group_index 两个自定义钩子函数, 是端到端测试的基础设施。
- tests/test_qwen2.5_0.5B_fanout_short.py (模块 fanout 测试; 类别 test; 类型 test-coverage; 符号 prepare, execute): 端到端测试文件, 验证完整 fanout 训练链路。

测试使用真实模型和数据集，通过自定义钩子检查 rollout_id 共享和分组归一化正确性。

- .github/workflows/pr-test.yml (模块 CI 配置; 类别 infra; 类型 infrastructure) : CI 工作流配置, 在 e2e-test-sglang 矩阵中新增本测试条目, 确保每次提交都自动运行。
- pyproject.toml (模块 项目配置; 类别 config; 类型 configuration) : 修复 ruff 配置格式, 添加 [tool.ruff.lint] section 以适配新版 ruff。
- .github/workflows/pr-test.yml.j2 (模块 CI 模板; 类别 infra; 类型 infrastructure) : CI 工作流模板, 与 pr-test.yml 同步修改, 确保生成正确。

关键符号: compact_generate, grpo_normalize_by_group_index, prepare, execute

关键源码片段

slime/rollout/_fanout_test_helpers.py

核心辅助模块, 提供 compact_generate 和 grpo_normalize_by_group_index 两个自定义钩子函数, 是端到端测试的基础设施。

```
import copy
import os
from collections import defaultdict

MAX_FANOUT = 3
COUNTER_FILE_ENV = "SLIME_FANOUT_TEST_COUNTER_FILE"

async def compact_generate(args, sample, sampling_params):
    """One prompt to N siblings, deterministic N = 1 + (index % MAX_FANOUT).

    Strategy: call sglang once, deepcopy N-1 times. Bounded GPU cost —
    we're pinning the framework's per-rollout handling, not generation
    diversity.
    """
    from slime.rollout.sglang_rollout import generate

    # Counter file is used by the test to verify that every prompt
    # actually went through this function (no silent fallback).
    counter_path = os.environ.get(COUNTER_FILE_ENV)
    if counter_path:
        try:
            with open(counter_path, "a") as f:
                f.write(f"{sample.index}\n")
        except OSError:
            # best-effort
            pass

    base_sample = await generate(args, sample, sampling_params)
    # Number of siblings: 1, 2, or 3 — varies per prompt to stress uneven fanout.
    n = 1 + (sample.index % MAX_FANOUT)
    siblings = []
    for _ in range(n):
```

```

s = copy.deepcopy(base_sample)
# Critical: all siblings share rollout_id so the per-rollout reducer
# aggregates them as ONE rollout (not N) and the rollout-aware step
# splitter keeps them in the same step. group_index is inherited
# via deepcopy and is used by grpo_normalize_by_group_index below.
s.rollout_id = sample.index
siblings.append(s)
return siblings

```

tests/test_qwen2.5_0.5B_fanout_short.py

端到端测试文件，验证完整 fanout 训练链路。测试使用真实模型和数据集，通过自定义钩子检查 rollout_id 共享和分组归一化正确性。

```

import os
import tempfile
import slime.utils.external_utils.command_utils as U

TIGHT_DEVICE_MEMORY = U.get_bool_env_var("SLIME_TEST_TIGHT_DEVICE_MEMORY", "1")
MODEL_NAME = "Qwen2.5-0.5B-Instruct"
MODEL_TYPE = "qwen2.5-0.5B"
NUM_GPUS = 4
FANOUT_COUNTER_FILE = os.environ.get(
    "SLIME_FANOUT_TEST_COUNTER_FILE",
    os.path.join(tempfile.gettempdir(), "slime_fanout_test_counter.log"),
)

def prepare():
    """Download model and dataset, clear the test counter."""
    U.exec_command("mkdir -p /root/models /root/datasets")
    U.exec_command(f"hf download Qwen/{MODEL_NAME} --local-dir /root/models/{MODEL_NAME}")
    U.hf_download_dataset("zhuzilin/dapo-math-17k")
    # Ensure previous run invocations don't bleed into this test.
    try:
        os.remove(FANOUT_COUNTER_FILE)
    except FileNotFoundError:
        pass

def execute():
    """Run the training command with custom generate and reward post-process hooks."""
    ckpt_args = f"--hf-checkpoint /root/models/{MODEL_NAME}/ " f"--ref-load /root/models/{MODEL_NAME}/ "
    # Shape: rollout_batch_size=8, n_samples_per_prompt=1 (all fanout is
    # owned by compact_generate), global_batch_size=4, num_rollout=3.
    rollout_args = (
        "--prompt-data /root/datasets/dapo-math-17k/dapo-math-17k.jsonl "
        "--input-key prompt --label-key label --apply-chat-template "
        "--rollout-shuffle --rm-type deepscaler --num-rollout 3 "
        "--rollout-batch-size 8 --n-samples-per-prompt 1 "
    )

```

```
"--rollout-max-response-len 8192 --rollout-temperature 0.8 "  
"--global-batch-size 4 --balance-data "  
"--custom-generate-function-path slime.rollout._fanout_test_helpers.compact_generate "  
"--custom-reward-post-process-path slime.rollout._fanout_test_helpers.grpo_normalize_by_  
group_index "  
)  
# ... (train command and post-train counter assertion follow)
```

评论区精华

无 review 评论。但设计上有几点值得注意：辅助函数放在 `slime/rollout/` 而非 `tests/` 是因为 `--custom-generate-function-path` 通过 `importlib.import_module` 解析点路径，测试文件文件名包含点（`test_qwen2.5_0.5B_fanout_short.py`）无法被正确 import；因此采用 `underscored` 前缀的模块且放置在源码目录下。

- 暂无高价值评论线程

风险与影响

- 风险：风险较低。测试运行在 4 卡 GPU 上，增加了 CI 耗时；`pyproject.toml` 的 `ruff` 配置改动可能影响 linting 行为，但目前仅添加空 `section`，无实际影响。若测试在 CI 中不稳定（如 GPU 资源竞争或超时），可能需要调整超时设置。
- 影响：仅影响 CI 测试矩阵和开发调试流程。对用户功能无直接影响。但此测试覆盖了核心路径，能预防未来回归。
- 风险标记：CI 新增测试耗时，`ruff` 配置改动可能意外

关联脉络

- PR #1926 Move micro-batch scheduling from training side to rollout side: 引入了 rollout-aware 的 step splitter，本测试的 fanout 模式依赖此调度正确性。
- PR #1930 [1/N] Support training with variable global batch size: 支持动态 global batch size，fanout 中样本数变化依赖该基础设施。
- PR #1933 [2/N] Support training with variable global batch size: 继续 variable global batch size 工作，fanout 测试是对其的集成验证。
- PR #1921 Add example for streaming output: 展示了自定义 generate 函数的设计模式，本测试采用了类似方式。
- PR #1920 Move fully_async example to main codebase: 将自定义 rollout 模式纳入主代码库，fanout 测试是其延伸。