PR #2266 完整报告

THUDM/slime

Refactor --save-debug-train-data

合并时间：2026-08-12 17:50

原文链接：http://prhub.com.cn/THUDM/slime/pull/2266

执行摘要

- 一句话：重构 --save-debug-train-data 为单文件转储并支持 CP 恢复
- 推荐动作：值得精读。重点看 restore_context_parallel_fields_to_cpu 的逐 sample gather 与显存控制、policy_loss_function 中的捕获钩子设计，以及 _build_dump_payload 的排序与布局。对需要扩展训练数据转储或理解 Megatron 后端 CP/PP 数据流转的工程师有直接参考价值。

功能与动机

PR body 为空，动机可从代码注释和提交信息还原。旧实现每 rank 各写一个文件，样本顺序与 rollout dump 不对齐，无法逐样本比对；CP 开启时 response 级字段只有本 rank 局部片段，落盘数据不可直接用；当配置跳过单独 log-prob 重算（can_reuse_log_probs_in_loss / use_rollout_logprobs）时转储拿不到 per-sample log_probs，需要额外 forward。新实现希望在不多跑 forward、不显著抬高显存峰值的前提下，产出与 rollout dump 严格对齐的规范化 train dump。

实现拆解

1. 转储核心迁移与重建：新增 slime/backends/megatron_utils/train_dump_utils.py，删除 slime/utils/train_dump_utils.py。新的 save_debug_train_data 只由“最后一个 PP stage + TP rank 0 + CP rank 0”的 writer rank 写盘；_build_dump_payload 把 DP gather 到的各 rank 载荷整理成与 rollout dump 对齐的结构：samples 按 partition 优先排序、sample_indices 兜底；micro_batch_indices、num_microbatches、global_batch_sizes 等布局字段挂在 dp_shards 键下，避免复制 per-sample 张量。
2. CP 字段逐步恢复与显存控制：restore_context_parallel_fields_to_cpu 遍历 _CONTEXT_PARALLEL_FIELDS（共 10 个 response 级字段，如 rollout_log_probs、values、advantages、returns）逐 sample 调用 gather_tensor(value, total_length, response_length) 恢复完整序列；keep_restored 只对 writer rank 生效，其余 rank 跑完集合通信即丢弃结果，每次 gather 后立刻 del full_value，把额外设备显存峰值压到“一个 response 张量”以内。
3. 训练期 log-prob 捕获：slime/backends/megatron_utils/loss.py 新增模块级 _LOG_PROB_CAPTURE 及 enable_log_prob_capture、drain_captured_log_probs、_maybe_capture_log_probs；policy_loss_function 在 torch.cat 重绑定 log_probs 之前调用捕获钩子，按全局 partition 位置保存 CP-local 张量，“首次出现优先”保证多 step 训

练保留 old-policy 值。actor.py 的 train_actor 在开启 dump 且 rollout_data 缺少 log_probs 时启用捕获，train 结束后以本地 partition 顺序还原注入 rollout_data["log_probs"]。

4. 数据契约与配置联动：slime/utils/data.py 在 dump 模式下向训练数据注入 partition 键；model.py 配合格式版本控制；arguments.py 微调参数语义；dump 文件改为 format_version=2，单文件同时含 samples 和 dp_shards 两个顶层键。
5. 测试、CI 与文档：新增 tests/test_train_dump.py（真实集合通信的 CP 恢复、DP 分片单文件写入、无 CP 归一化格式）与 8 GPU *e2e* tests/test_qwen2.5_0.5B_debug_train_dump_e2e.py（TP=2、PP=2、CP=2、DP=1，按 rollout_position 连接两份 dump 并逐样本比对 rollout_log_probs）；pr-test.yml 及其模板注册新测试，docs/en/developer_guide/debug.md 补充新格式说明。

关键文件：

- slime/backends/megatron_utils/train_dump_utils.py（模块 转储工具；类别 source；类型 dependency-wiring；符号 _to_cpu, restore_context_parallel_fields_to_cpu, _is_per_sample, _build_dump_payload）：转储核心实现，从 slime/utils 迁入并重写为单文件格式；包含 CP 字段恢复、payload 构建与 writer 选择逻辑，是本 PR 的主体。
- slime/backends/megatron_utils/loss.py（模块 损失函数；类别 source；类型 core-logic；符号 enable_log_prob_capture, drain_captured_log_probs, _maybe_capture_log_probs）：在策略损失主路径新增可选 log-prob 捕获钩子，使 dump 在复用训练 forward 时零额外开销拿到 per-sample log-probs。
- slime/backends/megatron_utils/actor.py（模块 训练器；类别 source；类型 dependency-wiring；符号 train_actor）：接线点：根据 save_debug_train_data 与 rollout_data 是否含 log_probs 决定是否开启捕获，train 后按 partition 还原并注入。
- tests/test_train_dump.py（模块 单元测试；类别 test；类型 test-coverage；符号 _patch_single_dp_writer, _restore_context_parallel_worker, _single_file_dump_worker, test_restore_context_parallel_fields_with_real_collective）：覆盖新转储核心：真实集合通信 CP 恢复、DP 分片单文件写入、无 CP 归一化格式。
- tests/test_qwen2.5_0.5B_debug_train_dump_e2e.py（模块 端到端测试；类别 test；类型 test-coverage；符号 prepare, _train_args, _verify, execute）：8 GPU 端到端验证：TP/PP/CP 全开，连接 rollout 与 train dump 逐样本比对 rollout_log_probs。
- slime/utils/train_dump_utils.py（模块 转储工具；类别 source；类型 deletion；符号 save_debug_train_data）：旧实现被删除，逻辑迁移至 megatron_utils 下的新文件。

关键符号：save_debug_train_data, restore_context_parallel_fields_to_cpu, _build_dump_payload, _to_cpu, _is_per_sample, enable_log_prob_capture, drain_captured_log_probs, _maybe_capture_log_probs

关键源码片段

slime/backends/megatron_utils/loss.py

在策略损失主路径新增可选 log-prob 捕获钩子，使 dump 在复用训练 forward 时零额外开销拿到 per-sample log-probs。

```
# 训练 forward 中产生的 per-sample log-probs 快照，按全局 rollout position 索引。
# 只在需要 dump 且复用训练 forward 的 log-probs 时启用，避免额外 forward。
_LOG_PROB_CAPTURE: "dict[int, torch.Tensor] | None" = None


def enable_log_prob_capture() -> None:
    """开启捕获，必须在 train 之前调用。"""
    global _LOG_PROB_CAPTURE
    _LOG_PROB_CAPTURE = {}


def drain_captured_log_probs() -> "dict[int, torch.Tensor]":
    """取走捕获结果并停止捕获；未捕获时返回空 dict。"""
    global _LOG_PROB_CAPTURE
    captured = _LOG_PROB_CAPTURE or {}
    _LOG_PROB_CAPTURE = None
    return captured


def _maybe_capture_log_probs(batch: RolloutBatch, log_probs: list[torch.Tensor]) -> None:
    """在 policy_loss_function 中快照 per-sample CP-local log-probs。

    仅在捕获开启且 batch 携带 partition 时工作；partition 只在 dump 模式下
    被注入训练数据。每个 position 保留首次出现的值（old-policy），多 step
    训练因此不会覆盖初始策略的 log-probs。
    """
    if _LOG_PROB_CAPTURE is None:
        return
    positions = batch.get("partition")
    if not positions:
        return
    for position, log_prob in zip(positions, log_probs, strict=True):
        if position not in _LOG_PROB_CAPTURE:
            _LOG_PROB_CAPTURE[position] = log_prob.detach().clone()
```

评论区精华

本PR没有review评论或讨论线程（comments_count=0、review_comments_count=0）。设计取舍体现在代码注释与提交信息中：单文件格式中 dp_shards 布局单独存放，是为了与 rollout debug dump 的扁平 samples 视图对齐而不复制 per-sample 张量；writer 只由“last PP stage + TP0 + CP0”承担，e2e 测试专门覆盖该选取逻辑；keep_restored 仅在 writer 上保留 CPU 值并配合 del full_value 控制显存峰值，是最值得借鉴的写法。

- 暂无高价值评论线程

风险与影响

- 风险：

1. 核心训练路径改动：loss.py 的 policy_loss_function 是训练主路径，_maybe_capture_log_probs 的新分支必须保证默认关闭时零开销、零行为变化；注释明确要求钩子在 torch.cat 重绑定 log_probs 之前执行，后续维护需保持此顺序。
2. 多 rank 状态管理：模块级 _LOG_PROB_CAPTURE 依赖 actor.py 的对称启停；若 train 异常退出未调用 drain_captured_log_probs，可能残留旧状态，建议补充异常路径清理。
3. 新增 CP 集合通信：restore_context_parallel_fields_to_cpu 对每个 CP 字段逐个 sample gather，字段多时通信次数线性增加；虽有 del 控制显存，通信延迟对训练循环仍可能有扰动，目前仅在 dump 模式启用。
4. 数据格式契约变更：dump 产物从“每 rank 一个文件”变为“单文件 + format_version=2”，依赖旧格式的调试脚本需要迁移；partition 键只在 dump 模式注入，正常训练数据契约不变。
5. e2e 覆盖有限：8 GPU e2e 覆盖 TP=2/PP=2/CP=2/DP=1，未覆盖 DP>1 与 CP 同时存在的组合（单元测试用 4 进程模拟 DP=2/CP=2）。 - 影响：影响范围集中在 Megatron 后端训练调试链路：使用 --save-debug-train-data 的用户会拿到格式变更后的产物（单文件、format_version=2），需要相应调整调试脚本；新 log-prob 捕获仅在开启 dump 且复用训练 forward 时生效，对正常运行无感知；CI 增加 8 GPU e2e 测试成本；团队后续可以用同一份规范化 dump 直接对齐 rollout 与 train 数据，降低排查策略 / 奖励不一致的难度。 - 风险标记：核心训练路径改动，数据格式契约变更，新增 CP 集合通信，多 rank 状态管理

关联脉络

- PR #2247 fix: forward dual-clip PPO epsilon: 同改 slime/backends/megatron_utils/loss.py，都在训练损失主路径上做配置 / 数据接线。
- PR #2234 fix: pair --log-correct-samples rewards with the DP-local samples: 同改 slime/utils/data.py，都处理 DP 切分下样本与张量的对齐问题。
- PR #2205 perf: vectorize REINFORCE++ discounted returns: 同属训练工具链数值逻辑重构，并配套新增测试，反映该区域的持续演进。