

# PR #2123 完整报告

THUDM/slime

Fix training stuck on all-gather cp

合并时间: 2026-07-01 14:21

原文链接: <http://prhub.com.cn/THUDM/slime/pull/2123>

## 执行摘要

- 一句话: 修复 all-gather CP 模式下训练卡住的问题
- 推荐动作: 建议合并该 PR, 它解决了严重的训练卡死问题。修改小而精确, 值得精读以理解 all-gather CP 模式下梯度图构建的细节。特别是 `requires_grad` 继承和切片取代 `torch.zeros` 的做法, 可作为保持计算图连续性的参考模式。

## 功能与动机

在 all-gather context parallel 训练场景下, 模型会卡住停止训练。作者通过排查发现, `_allgather_cp_redistribute` 和 `_extract_per_sample` 中创建零张量的方式存在梯度图不一致问题, 以及 `policy_loss_function` 中 `logprob` 比较使用了错误的变量, 这些都会导致训练卡死。

## 实现拆解

1. `_allgather_cp_redistribute` 中零张量梯度传递修复 (`loss.py:202-205`): 将 `torch.zeros(response_length, ... requires_grad=True)` 改为 `requires_grad=ref_value.requires_grad`, 确保零张量的 `requires_grad` 属性与原始参考值一致, 避免在特定 CP 分区下梯度图断裂。
2. `_extract_per_sample` 中空切片创建方式优化 (`loss.py:444-449`): 将 `torch.zeros((0), ..)` 替换为切片操作 `log_prob_full[:0]` 和 `entropy_full[:0]`, 保持与已有计算图的一致性和连续性, 消除不必要的 tensor 创建。
3. `policy_loss_function` 中 `logprob` 对比变量修复 (`loss.py:1073-1075`): 新增 `use_rollout_logprobs` 配置 (`args.use_rollout_logprobs`), 当为 `True` 时使用 `old_log_probs`, 否则使用当前 `log_probs`, 避免统一使用 `old_log_probs` 导致不匹配和梯度问题。

关键文件:

- `slime/backends/megatron_utils/loss.py` (模块 损失计算; 类别 source; 类型 core-logic; 符号 `_allgather_cp_redistribute`, `_extract_per_sample`, `policy_loss_function`): 核心修改文件, 包含三处关键 bug 修复: 梯度图连续性修复、空切片创建优化、`logprob` 对比逻辑修复。

关键符号: `_allgather_cp_redistribute`, `_extract_per_sample`, `policy_loss_function`

## 关键源码片段

## slime/backends/megatron\_utils/loss.py

核心修改文件，包含三处关键 bug 修复：梯度图连续性修复、空切片创建优化、logprob 对比逻辑修复。

```
# _allgather_cp_redistribute 中零张量 requires_grad 继承修复
if value is None or e <= s:
    full_resp = torch.zeros(
        response_length,
        dtype=ref_dtype,
        device=ref_device,
        requires_grad=ref_value.requires_grad, # 原为 True, 改为继承 ref_value
    )

# _extract_per_sample 中空切片方法优化
if e <= s:
    log_probs_list.append(log_prob_full[:0]) # 原为 torch.zeros((0,), ...)
    if entropy_full is not None:
        entropy_list.append(entropy_full[:0]) # 原为 torch.zeros((0,), ...)

# policy_loss_function 中 logprob 对比变量的选择
log_probs_to_compare = log_probs if args.use_rollout_logprobs else old_log_probs
# 根据配置选择正确的 log_probs, 避免错误变量导致梯度不一致
train_rollout_logprob_abs_diff = sum_of_sample_mean(
    (log_probs_to_compare - rollout_log_probs).abs()
)
```

## 评论区精华

该 PR 没有评论或 review 讨论。

- 暂无高价值评论线程

## 风险与影响

- 风险：风险较低：三个修改都是针对特定分支场景的边界条件修复，且修改范围小（只有 6 行增加、5 行删除）。主要风险在于 use\_rollout\_logprobs 配置若未正确设置，可能影响 rollout logprob 对比的计算逻辑，但不会导致崩溃。需要确保该配置在实际训练脚本中正确传递。
- 影响：直接影响所有使用 Megatron 后端且启用 all-gather context parallel 的训练任务，修复了训练卡死的问题，提升了训练稳定性。对非 all-gather CP 模式无影响。use\_rollout\_logprobs 配置的引入增加了灵活性，但需要用户明确设置。
- 风险标记：核心路径变更，配置键新增

## 关联脉络

- PR #2152 Optimize memory usage for \_VocabParallelLogProbEntropy: 同一文件（loss.py）之前的优化，涉及 same area 的修改。

- PR #2153 bugfix: 也修改了 loss.py 中 with\_entropy\_grad 相关逻辑, 属于同一模块的 bugfix。
- PR #2144 perf: fuse PPO logprob entropy computation: 涉及 loss.py 中 logprob entropy 计算的融合, 与本 PR 的 all-gather CP 逻辑相关。