

PR #6846 完整报告

verl-project/verl

[fsdp] fix: re-wrap fused log_probs/entropy when `use\_remove\_padding=False`

合并时间: 2026-06-26 17:03

原文链接: <http://prhub.com.cn/verl-project/verl/pull/6846>

执行摘要

- 一句话: 修复 fused kernel 在 no padding 模式下的布局不匹配崩溃
- 推荐动作: 建议精读以下两点: (1) 理解 fused kernel 输出与 no padding 模式下 nested tensor 布局的适配逻辑; (2) 注意 boolean mask 实现相对于 narrow+unbind+cat 的性能优势, 可推广到类似场景。

功能与动机

关联 Issue #6780 中报告了 on-policy 模式下的训练崩溃。根本原因是 `use_fused_kernels=True` 与 `use_remove_padding=False` 组合时, fused 分支返回稠密张量而非嵌套张量, 触发了 `padding.py:131` 的 `assert sequence_offsets[-1].item() == values.shape[0]` 错误。

实现拆解

1. 移除原 fused 分支错误实现: 在 `verl/workers/engine/fsdp/transformer_impl.py` 的 `else # not using rmpad` 块中, 删除了旧的 `response_length = ...` 行及直接切片赋值 `log_probs = output.log_probs[:, -response_length - 1 : -1]`, 该切片无法处理变长序列的 padding 对齐。
2. 针对 `NO_PADDING` 模式新增 fused 重包裹逻辑: 当 `pad_mode == DatasetPadMode.NO_PADDING` 时, 通过 `input_ids.offsets()` 获取 `cu_seqlens` 并计算每个序列的真实长度 `seq_lengths`, 然后使用 `boolean mask arange < seq_lengths.unsqueeze(1)` 从稠密 `output.log_probs` 和 `output.entropy` 中筛选出有效 token (`(total_nnz,)`), 最后通过 `nested_tensor_from_jagged` 将其转换为嵌套张量。该方案避免了 `narrow + unbind + cat` 的 Python 开销, 性能更优。
3. 新增异常处理: 对于其他不支持的非 `NO_PADDING` 的 `pad_mode`, 显式抛出 `NotImplementedError`, 避免静默错误。
4. 测试验证: PR body 中提供了端到端 GRPO 复现脚本, 验证了修复后 step 1 正常完成, 且不影响已有组合 (`use_remove_padding=True` 或 `use_fused_kernels=False`)。

关键文件:

- `verl/workers/engine/fsdp/transformer_impl.py` (模块 训练引擎; 类别 source; 类型 core-logic; 符号 `prepare_model_outputs`): 核心修复文件: 在 `prepare_model_outputs` 方法中增加了 fused kernel 分支对 `NO_PADDING` 模式的支持, 通过 `boolean mask` 将稠

密输出重包裹为嵌套张量，并添加了异常处理。

关键符号: `prepare_model_outputs`

关键源码片段

`verl/workers/engine/fsdp/transformer_impl.py`

核心修复文件: 在 `prepare_model_outputs` 方法中增加了 fused kernel 分支对 `NO_PADDING` 模式的支持, 通过 boolean mask 将稠密输出重包裹为嵌套张量, 并添加了异常处理。

```
def prepare_model_outputs(self, output, output_args, micro_batch, ...):
    # ... 前面的代码处理 use_remove_padding 和 ulysses sp 分支 ...
    else: # not using rmpad and no ulysses sp
        if use_fused_kernels:
            if pad_mode == DatasetPadMode.NO_PADDING:
                # Re-wrap dense fused (bsz, seqlen) outputs as nested/jagged:
                # mask out the padding columns per sequence and pack the valid tokens to (total_nnz,).
                cu_seqlens = input_ids.offsets()
                # seq_lengths[i] = 真实长度 (不含 padding)
                seq_lengths = cu_seqlens.diff()
                # arange: 0, 1, ..., seqlen-1
                arange = torch.arange(output.log_probs.shape[1], device=output.log_probs.device)
                # mask[i, j] = True 当且仅当 j < seq_lengths[i]
                mask = arange < seq_lengths.unsqueeze(1)

                # 用 boolean mask 扁平化有效 token, 无需 narrow/unbind/cat
                log_probs = output.log_probs[mask]
                log_probs = torch.nested.nested_tensor_from_jagged(log_probs, cu_seqlens)

                if calculate_entropy:
                    entropy = output.entropy[mask]
                    entropy = torch.nested.nested_tensor_from_jagged(entropy, cu_seqlens)
            else:
                # 其他 pad_mode 暂不支持 fused kernel
                raise NotImplementedError(f"pad_mode {pad_mode} not implemented")
        else:
            # 非 fused 分支: 原来的 logits 计算 + narrow/unbind/cat 逻辑
            # ...
```

评论区精华

gemini-code-assist[bot] 的 review 指出原实现 (已修改前) 使用 `narrow + unbind + cat` 存在 Python 开销和多次小分配问题, 建议使用 boolean mask 直接扁平化有效元素。该建议被采纳并体现在最终实现中。wuxibin89 批准了该 PR。

- 使用 boolean mask 替代 narrow+unbind+cat 优化性能 (performance): 作者采纳建议, 最终实现使用 boolean mask + 扁平化 + nested_tensor_from_jagged。

风险与影响

- 风险:

1. 回归风险: 低。修改仅影响 `use_fused_kernels=True` 且 `use_remove_padding=False` 的路径, 该路径此前直接崩溃, 修复后与其他路径 (非 `fused`、`remove padding`) 行为对齐。
2. 性能风险: 低。`boolean mask` 方案相比原 `narrow+unbind+cat` 更高效。
3. 兼容性风险: 无。不涉及配置 /API 变更。
4. 测试覆盖: 缺少自动化单元测试, 仅通过手动端到端训练验证。 - 影响: 影响范围: 修复了 `use_fused_kernels=True + use_remove_padding=False` 组合下的训练崩溃, 使该配置 (常见于 Qwen3.5 等长序列模型) 恢复正常运行。影响程度: 中。修复了关键路径上的确定性崩溃。用户: 使用 `fused kernel` 且未开启 `remove padding` 的用户将不再遇到断言错误。 - 风险标记: 核心路径变更, 缺少自动化测试覆盖

关联脉络

- PR #6780 [fully_async_policy] On-policy mode (trigger=1, staleness=0) didn't match main_ppo training - converges much faster: 本 PR 直接修复了该 issue 中报告的崩溃 (`fused kernel + no padding` 组合)。
- PR #6777 [hardware] refactor: per-model NPU patches with fault isolation: 同仓库近期对 FSDP 引擎的修改, 涉及同一文件 `transformer_impl.py` 的兼容性调整。
- PR #6512 [fsdp, model] feat: per-unit LoRA summon, FSDP1/2 compatibility, and strip-modules support: 同仓库近期对 FSDP 引擎的较大重构, 涉及同一文件。