PR #6836 完整报告

verl-project/verl

[megatron] fix: return 3-tuple under calculate_per_token_loss to fix MoE aux/z-loss grad blowup at CP>1

合并时间：2026-07-13 10:57

原文链接：http://prhub.com.cn/verl-project/verl/pull/6836

执行摘要

- 一句话：修复 Megatron CP>1 时 MoE aux/z-loss 梯度爆炸
- 推荐动作：强烈建议使用 Megatron MoE 且 CP>1 的用户合入此 PR。值得关注的设计决策：精确计算路由 token 数的 _routed_num_tokens 方法、通过预乘后除抵消的二阶段归一化策略、以及对不合理配置的主动防护。

功能与动机

当使用 Megatron 后端且 CP>1 时，部分模型（如 Qwen3-VL）通过 Megatron-Bridge 自动启用 calculate_per_token_loss=True。该模式下 loss 函数需返回 (loss_sum, num_tokens, output) 三要素以便归一化，但 verl 始终返回 2 元组，导致 finalize_model_grads 跳过归一化，MoE aux/z-loss 梯度被放大数千倍（Issue #6609）。

实现拆解

1. 新增 _routed_num_tokens 方法（verl/workers/engine/megatron/transformer_impl.py）：根据 attention_mask 或 input_ids 计算实际路由 token 数（非填充）。
2. 在 forward_backward_batch 中计算全局路由 token 数：当 calculate_per_token_loss 启用时，调用 _routed_num_tokens 并通过 DP 组 all-reduce 得到全局值，存入 data 元数据。
3. 修改 postprocess_micro_batch_func 返回 3 元组：将 loss 预乘 routed_num_tokens / dp_size，使 Megatron 的最终除法抵消；返回 (loss, local_num_tokens, output)。
4. 添加安全守卫：拒绝 loss_agg_mode='seq-mean-token-mean' 和 use_remove_padding=False 组合，因无法正确抵消。
5. 回归测试（tests/models/test_moe_zloss_per_token_loss.py）：对比标志开启前后的梯度范数，确保不变。

关键文件：

- verl/workers/engine/megatron/transformer_impl.py（模块 引擎层；类别 source；类型 core-logic；符号 _routed_num_tokens, forward_backward_batch, postprocess_micro_batch_func）：核心修复：新增 _routed_num_tokens 方法，修改 forward_backward_batch 和 postprocess_micro_batch_func 以支持 per-token loss 三要素返回。

- tests/models/test_moe_zloss_per_token_loss.py（模块 模型测试；类别 test；类型 test-coverage；符号 _create_tiny_moe_model, _build_data, _grad_norm, test_moe_zloss_invariant_to_per_token_loss）：新增回归测试，验证 calculate_per_token_loss 标志不影响 MoE z-loss 梯度范数，覆盖核心修复。

关键符号：_routed_num_tokens, forward_backward_batch, postprocess_micro_batch_func, test_moe_zloss_invariant_to_per_token_loss

关键源码片段

verl/workers/engine/megatron/transformer_impl.py

核心修复：新增 _routed_num_tokens 方法，修改 forward_backward_batch 和 postprocess_micro_batch_func 以支持 per-token loss 三要素返回。

```
def _routed_num_tokens(self, data: TensorDict) -> torch.Tensor:
    # 实际输入 MoE router 的 token 数（非填充）
    # 优先使用 attention_mask（RL 路径），回退到 input_ids 计数（SFT / no-padding）
    attention_mask = data.get("attention_mask", None)
    if attention_mask is not None:
        return attention_mask.sum()
    input_ids = data["input_ids"]
    if input_ids.is_nested:
        return input_ids.offsets()[-1]
    return torch.tensor(input_ids.numel(), device=input_ids.device)

# 在 forward_backward_batch 中，于 batch_num_tokens 计算后插入：
if self.tf_config is not None and self.tf_config.calculate_per_token_loss:
    routed_num_tokens = self._routed_num_tokens(data).to(get_device_id())
    torch.distributed.all_reduce(
        routed_num_tokens, op=torch.distributed.ReduceOp.SUM,
        group=self.get_data_parallel_group()
    )
    tu.assign_non_tensor(data, routed_num_tokens=routed_num_tokens.item())
```

评论区精华

gemini-code-assist指出 forward_backward_batch 中直接索引 data["attention_mask"] 可能导致 KeyError（SFT 路径仅有 response_mask），建议改安全访问。EricMarcus-ai采纳并修复，同时后续发现并修正了 cp_size 叠加的归一化 bug。HollowMan6LGTM 并请 wuxibin89 复核，最终合并。

- attention_mask fallback (correctness): 作者接受建议，改为 data.get("attention_mask", None)，并在后续 commit 中修复了 cp_size 归一化问题。

风险与影响

- 风险：
  1. BSHD 路径（use_remove_padding=False）被显式禁止，可能影响已用此配置的用户。

2. loss_agg_mode='seq-mean-token-mean' 被拒绝，用户需改用 token-mean 等默认选项。
3. 新增 GPU 回归测试需 Ray 和 tokenizer，可能未纳入 CI 导致长期覆盖不足。
4. 仅修复 Megatron 后端，FSDP 等其他后端无影响。 - 影响：直接影响所有使用 Megatron 引擎 + MoE + CP>1 的训练任务（如 Qwen3-MoE 系列），修复后梯度范数恢复合理范围，训练稳定收敛。对 CP=1 或 FSDP 后端无影响。用户之前可能因梯度爆炸而被迫关闭 aux/z-loss 或使用 CP=1，合入后即可启用。 - 风险标记：BSHD+CP 被禁止，loss_agg_mode 受限，GPU 测试未入 CI

关联脉络

- 暂无明显关联 PR