PR #5942 完整报告

verl-project/verl

Revert "[megatron] fix: Adjust the attention mask shape for VLM with Megatron on NPU"

合并时间：2026-04-09 16:08

原文链接：http://prhub.com.cn/verl-project/verl/pull/5942

执行摘要

本 PR 回滚了 PR#5904 中对 VLM+Megatron 在 NPU 上注意力掩码形状的修复，将相关逻辑内联到模型前向函数中并移除了 NPU 专用处理。这是一个紧急回滚操作，可能因原始修复存在问题而触发，直接影响 NPU 环境下视觉语言模型的训练兼容性，建议团队关注后续替代方案。

功能与动机

PR body 仅简短说明“Reverts verl-project/verl#5904”，未提供具体回滚原因。结合 PR#5904 的动机（为 VLM+Megatron 在 NPU 环境适配注意力掩码形状）推测，原始修复可能引入了未预料的问题或未被充分验证，需要暂时回退到更稳定的代码状态以保障系统可靠性。

实现拆解

主要变更集中在两个文件：

1. verl/models/mcore/model_forward.py：在 gptmodel_forward_model_engine 函数中，将 PR#5904 新增的 VLM 注意力掩码构建逻辑内联，但移除了 NPU 相关的特殊处理（即不再为 NPU 构建掩码）。关键变更包括：
   - 移除对 build_vlm_attn_mask_thd 和 build_vlm_attn_mask_bshd 的调用。
   - 内联序列长度对齐和掩码创建循环（例如：for i, seqlen in enumerate(seqlens_in_batch): attention_mask[i, :seqlen] = True）。
   - 在 NPU 环境下，attention_mask 可能返回 None 或沿用默认逻辑。
2. verl/models/mcore/util.py：
   - 删除 _build_npu_attn_mask 函数（原用于构建 NPU 融合注意力所需的 B1SS 格式掩码）。
   - 移除 build_vlm_attn_mask_thd 和 build_vlm_attn_mask_bshd 两个辅助函数。
   - 调整 preprocess_bshd_engine 和 postprocess_bshd_engine 中的掩码处理逻辑，简化 NPU 路径。

评论区精华

Review 中仅有一个来自 gemini-code-assist[bot] 的自动化评论，针对内联的掩码创建循环提出性能优化建议：

> “The for-loop to create the attention mask can be vectorized for better performance, especially with larger batch sizes. You can replace the loop with a

> more efficient broadcasting operation."

建议使用 `torch.arange(...)[None, :] < seqlens_in_batch[:, None]` 替代循环。该建议未被作者回应或采纳，PR 直接合并，表明这可能是一个紧急回滚，优先考虑功能恢复而非性能优化。

风险与影响

- 功能风险：回滚后，VLM+Megatron 在 NPU 上可能无法正确处理注意力掩码，导致训练错误或性能下降，因为原始修复正是为了解决 NPU 适配问题。
- 维护风险：内联逻辑增加了 gptmodel_forward_model_engine 函数的复杂度，降低了代码可读性和可测试性。
- 性能风险：内联的循环掩码创建方式（如 review 所指）可能比向量化操作效率低，尤其在批量大时影响吞吐。
- 影响范围：主要影响使用 VLM+Megatron 在 NPU 上进行强化学习训练的用户，可能需要临时规避或等待后续修复。

关联脉络

- 直接关联：本 PR 是 PR#5904 的回滚，两者涉及相同文件（model_forward.py 和 util.py）和相同功能（VLM 注意力掩码处理）。PR#5904 原本是为了解决 Issue#5878 中报告的 NPU 兼容性问题。
- 技术脉络：近期多个 PR（如 #5909、#5680）都涉及 Megatron 后端和 NPU 平台的适配优化，表明团队正持续投入提升 Ascend 硬件上的训练能力。本回滚可能反映了 NPU 生态中视觉语言模型支持的复杂性，需要更稳健的解决方案。
- 演进趋势：从 PR#5904 的修复到本 PR 的回滚，显示出在快速迭代中，对底层模型引擎的变更需要更充分的测试和验证，尤其是在跨平台（NPU vs GPU）场景下。