

PR #6206 完整报告

verl-project/verl

[megatron] fix: fix seq_len pad len, and adapt to new mtp_loss api (for megatron dev brance)

合并时间: 2026-05-06 11:57

原文链接: <http://prhub.com.cn/verl-project/verl/pull/6206>

执行摘要

- 一句话: 修复 Megatron CP+TP 序列对齐与新 MTP API 适配
- 推荐动作: 该 PR 修复了重要的并行对齐 bug 并适配了新 API, 值得对 Megatron 引擎开发者和经验丰富的用户精读。关键设计决策: 使用条件导入实现 API 版本兼容, 以及对齐公式背后的推导逻辑 (乘积而非 lcm) 值得记录。建议为修改的路径补充示例或测试。

功能与动机

根据 PR body 描述, 需要修复两个问题: 1) 当同时启用 CP 和 TP 时, seq_len 应该填充到 $(2 * cp) * tp$; 2) 适配新的 mtp_loss API 以兼容 Megatron dev 分支。这些修复确保在 CP+TP 配置下序列填充正确, 并保持与新版 Megatron MTP 计算接口的兼容性。

实现拆解

1. 序列长度对齐修正: 在 verl/models/mcore/util.py 的 preprocess_bshd_engine 和 build_vlm_attn_mask_bshd 函数中, 将对齐规则从 $\text{math.lcm}(tp_size, 2 * cp_size)$ 改为 $tp_size * cp_size * 2$, 并更新注释说明原因: zigzag CP 下每 rank 持有 seq_len / cp_size 个 token, 之后还需被 tp_size 整除以支持 Sequence Parallel 的 scatter 操作。
2. 新版 MTP API 适配: 在 verl/models/mcore/mtp_patch.py 中, 通过 try/except 导入 process_mtp_loss 并定义 _HAS_PROCESS_MTP_LOSS 标志。在 _megatron_gptmodel_postprocess 函数中, 当 _HAS_PROCESS_MTP_LOSS 为 True 时, 调用 _process_mtp_loss 一次性完成 hidden_states 的 chunk、rolling、loss scaling 和输出层计算, 替代原先的手动循环。同时为函数签名新增 is_spec_decode 参数以对齐新版接口。
3. 向后兼容保留: 若导入失败 (旧版 Megatron), _HAS_PROCESS_MTP_LOSS 为 False, 继续执行原有的逐层 MTP 损失计算逻辑, 不破坏现有功能。
4. 辅助调整: 修正 cp_group 的获取方式, 优先从 self.cp_group 获取, 否则回退到 self.pg_collection.cp (若存在)。

关键文件:

- verl/models/mcore/mtp_patch.py (模块 模型层; 类别 source; 类型 data-contract; 符号 _HAS_PROCESS_MTP_LOSS, _process_mtp_loss, _megatron_gptmodel_postprocess, patch_postprocess): 核心文件: 适配新版 Megatron MTP 损失 API, 增加新 / 旧 API 自动选择路径, 调整函数签名。

- verl/models/mcore/util.py (模块 模型层; 类别 source; 类型 data-contract; 符号 preprocess_bshd_engine, build_vlm_attn_mask_bshd) : 修改序列长度对齐规则, 保证 CP+TP 配置下填充正确。

关键符号: _megatron_gptmodel_postprocess, preprocess_bshd_engine, build_vlm_attn_mask_bshd

关键源码片段

verl/models/mcore/mtp_patch.py

核心文件: 适配新版 Megatron MTP 损失 API, 增加新 / 旧 API 自动选择路径, 调整函数签名。

```
# verl/models/mcore/mtp_patch.py (head)
# 从 Megatron 尝试导入新版 process_mtp_loss
try:
    from megatron.core.transformer.multi_token_prediction import (
        process_mtp_loss as _process_mtp_loss
    )
    _HAS_PROCESS_MTP_LOSS = True
except ImportError:
    _HAS_PROCESS_MTP_LOSS = False

# ... 省略其它导入和辅助函数 ...

def _megatron_gptmodel_postprocess(
    self,
    hidden_states,
    input_ids,
    # ... 其它参数 ...
    is_spec_decode=None, # 新增参数, 对齐新版 Megatron 接口
):
    # ... 前置逻辑 ...

    if self.config.mtp_num_layers and labels is not None:
        if _HAS_PROCESS_MTP_LOSS:
            # 新版 API 路径: 一次调用完成 chunk、roll、loss 计算及缩放
            # 获取 cp_group (优先使用属性, 否则从 pg_collection 回退)
            cp_group = getattr(self, "cp_group", None) or (
                self.pg_collection.cp if hasattr(self, "pg_collection") else None
            )
            # 如果需要  $\mu$ P 缩放
            scale_logits_fn = (
                self._scale_logits
                if (hasattr(self, "_scale_logits") and self.config.use_mup)
                else None
            )
            hidden_states = _process_mtp_loss(
                hidden_states=hidden_states,
```

```
        labels=labels,
        loss_mask=loss_mask,
        output_layer=self.output_layer,
        output_weight=output_weight,
        runtime_gather_output=runtime_gather_output,
        is_training=self.training,
        compute_language_model_loss=self.compute_language_model_loss,
        config=self.config,
        cp_group=cp_group,
        packed_seq_params=packed_seq_params,
        scale_logits_fn=scale_logits_fn,
    )
else:
    # 旧版 API 路径（保留原始逐层手动计算）
    # 此处在 head 中未展示，但功能保持原状
    pass
# ... 后续 logits 及 loss 计算 ...
```

评论区精华

本 PR 无实质 review 讨论。Gemini-code-assist[bot] 仅做了评论摘要并声明无反馈，维护者 wuxibin89 直接批准合并。

- 暂无高价值评论线程

风险与影响

- 风险：

1. 向后兼容风险：新 API 路径通过 try/except 保护，旧路径保留，风险可控。但若新版 Megatron 的 process_mtp_loss 接口签名与预期不符，可能引发运行时错误。
2. 序列对齐变更风险：对齐从 lcm(tp_size, 2*cp_size) 改为 tp_size * cp_size * 2，在 cp=1 时结果相同（均为 tp_size），但 cp>1 时新值更严格（可能更大），可能导致更多的 padding 和显存占用，但不会出错。
3. 缺少测试覆盖：未新增单元测试或 E2E 测试验证新 API 路径，存在回归隐患。
4. 参数变更：函数签名增加 is_spec_decode 参数，调用处若未更新可能遗漏参数（但 patch 可见调用处已处理）。- 影响：直接影响使用 Megatron 引擎并启用 Multi-Token Prediction (MTP) 的用户，特别是同时使用 CP 和 TP 的用户。修复了之前因对齐不足可能导致的隐藏状态错位或计算错误，不影响纯 TP 或纯 CP 场景。对新版 Megatron dev 分支的用户，PR 提供了 MTP 计算路径的平滑迁移。整体影响范围较小，但正确性关键。- 风险标记：核心路径变更，缺少测试覆盖，向后兼容风险

关联脉络

- 暂无明显关联 PR