PR #7593 完整报告

verl-project/verl

[trainer] fix: make synthetic padding safe for context parallelism

合并时间：2026-08-28 15:26

原文链接：http://prhub.com.cn/verl-project/verl/pull/7593

执行摘要

- 一句话：增大合成 padding 序列长度，适配上下文并行
- 推荐动作：本 PR 是关键的 bugfix，建议阅读 padding_utils.py 中的相关实现，理解为何较小的 padding 序列在 CP 下会导致问题。关注其未配套测试的风险，可在后续 PR 中补充覆盖不同 CP 大小的测试用例。

功能与动机

PR body 中指出：在 packed context-parallel 预处理中，当 TP × CP 配置增大时，原先仅包含 1 个 prompt token 的输入序列可能导致 source slice 为空，从而引发错误。将 padding 序列长度提升到 128 个 attention-valid token，可确保在 TP × CP 配置高达 64 时不会出现空窗口，保证训练稳定。

实现拆解

1. 定义常量 SYNTHETIC_PADDING_SEQ_LEN = 128，替换原先硬编码的序列长度。
2. 修改 construct_minimal_padding_template 函数：将 prompts 由 1 个 token 扩展为 SYNTHETIC_PADDING_SEQ_LEN - 1 个 token（127），responses 保持 1 个 token，input_ids 通过 torch.cat 拼接 prompts 与 responses，形成总长 128 的序列。
3. 相应调整 attention_mask、response_mask 等张量的长度计算，response_mask 仍只对 response 位置置 0，保持 loss masking 行为不变。
4. 更新 template_tag 中的 prompt_len、response_len、seq_len 字段，使其与新序列长度一致，确保下游指标计算和批次划分正确。
5. 更新 upsample_batch_to_divisible_size 的 docstring，说明新的 padding 长度。
6. 未伴随新增测试，需要人工验证或后续补充。

关键文件：

- verl/trainer/ppo/padding_utils.py（模块 训练器；类别 source；类型 core-logic；符号 construct_minimal_padding_template, upsample_batch_to_divisible_size）：本次 PR 的唯一变更文件，修改了合成 padding 样本的构造逻辑，是修复的核心。

关键符号：construct_minimal_padding_template, upsample_batch_to_divisible_size

关键源码片段

verl/trainer/ppo/padding_utils.py

本次 PR 的唯一变更文件，修改了合成 padding 样本的构造逻辑，是修复的核心。

```
# verl/trainer/ppo/padding_utils.py

# 常量定义：合成 padding 序列总长度（attention-valid token 数）
# 该值需足够大，以确保在 packed context-parallel 预处理时不会出现空 source slice
SYNTHETIC_PADDING_SEQ_LEN = 128


def construct_minimal_padding_template(
    source_td: dict,
    source_tag: dict,
    eos_token_id: int,
) -> tuple[dict, dict]:
    """Construct a text-only padding template.

    Args:
        source_td: A single sample dict retrieved from TransferQueue.
        source_tag: The corresponding tag dict for that sample.
        eos_token_id: The EOS token id from the tokenizer.

    Returns:
        A tuple of (template_sample, template_tag) ready for padding.
    """
    # Copy the sample template from an existing sample.
    template_sample = {}
    for key in source_td.keys():
        value = source_td[key]
        template_sample[key] = value.clone() if isinstance(value, torch.Tensor) else copy.
        deepcopy(value)

    # Deep copy the template tag from an existing sample.
    template_tag = copy.deepcopy(source_tag)

    # Build minimal sequence：prompt 部分填充 127 个 EOS token，response 部分 1 个 EOS
    token
    # 总长 128，确保 TP × CP 配置下不会出现空窗口
    responses = torch.full((1,), eos_token_id, dtype=torch.int64)
    prompts = torch.full((SYNTHETIC_PADDING_SEQ_LEN - 1,), eos_token_id, dtype=torch.int64)
    input_ids = torch.cat((prompts, responses))
    attention_mask = torch.ones_like(input_ids, dtype=torch.int64)
    # response 部分完全屏蔽 loss，避免影响训练
    response_mask = torch.zeros_like(responses)
    position_ids = build_padding_position_ids(template_sample.get("position_ids"), attention_mask)
    routed_experts = build_padding_routed_experts(template_sample.get("routed_experts"), input_
    ids.size(0))

    # Update the fields and remove redundant parts
```

```
template_sample.update(
    prompts=prompts,
    responses=responses,
    input_ids=input_ids,
    attention_mask=attention_mask,
    position_ids=position_ids,
    num_turns=0,
    response_mask=response_mask,
    loss_mask=response_mask,
    rm_scores=torch.zeros_like(response_mask, dtype=torch.float32),
    rollout_log_probs=torch.zeros_like(response_mask, dtype=torch.float32),
)
if "multi_modal_inputs" in template_sample:
    template_sample["multi_modal_inputs"] = {}
if routed_experts is not None:
    template_sample["routed_experts"] = routed_experts
else:
    template_sample.pop("routed_experts", None)

# Padding flag is deployed to protect metrics calculation (e.g. response length, score,
reward).
# 同步更新长度字段，保证下游指标过滤和 batch 划分正确
template_tag.update(
    is_padding=True,
    prompt_len=SYNTHETIC_PADDING_SEQ_LEN - 1,
    response_len=1,
    seq_len=SYNTHETIC_PADDING_SEQ_LEN,
)
return template_sample, template_tag
```

评论区精华

本 PR 无 review 评论，仅有 maintainer wuxibin89 的 APPROVED 审核，无讨论线程。

- 暂无高价值评论线程

风险与影响

- 风险：
  1. 序列长度显著增大（2→128）可能导致合成 padding 样本的显存占用和计算量增加，但合成样本数量通常较少，影响相对有限。
  2. 修改了 padding 序列结构，需要确保后续所有依赖 prompt_len 和 response_len 的逻辑（如指标计算、batch 划分）能正确处理新的长度，存在回归风险。
  3. 未补充单元测试，可能无法覆盖所有 TP × CP 配置和边界情况。
  4. 由于没有 review 讨论，实现细节和潜在问题可能未被充分审视。 - 影响：影响范围：verl trainer 的 padding 逻辑，主要影响使用 context parallelism（CP）的配置，特别是 TP × CP 较大的场景。对用户而言，修复了可能在训练启动或运行时报错的问题，提

升了训练的稳定性。对系统而言，增加了合成序列的长度，可能轻微增加计算和显存开销。
- 风险标记：核心路径变更，缺少测试覆盖

关联脉络

- PR #7553 [trainer] fix: *enforce* strict dynamic micro-batch token limits: 同样涉及 trainer 的 batch 处理和 token 长度限制，可能与本 PR 的 padding 逻辑相关。
- PR #7562 [data] fix: offload multimodal dataset processing: 涉及数据预处理，可能影响 padding 相关的数据构造。