PR #6268 完整报告

verl-project/verl

[model, fsdp] fix: honor SP-rolled labels in fused kernels (#6068)

合并时间：2026-05-14 08:27

原文链接：http://prhub.com.cn/verl-project/verl/pull/6268

执行摘要

- 一句话：修复 fused kernels 在 Ulysses SP 下的标签滚动错误
- 推荐动作：此 PR 是重要的正确性修复，建议所有使用 fused kernels + Ulysses SP 的团队尽快合并并同步至下游分支。同时值得关注的设计决策是：由 engine 统一预滚动标签（而非在 fused forward 中重新计算），保持了单 SP 和多 SP 路径的一致性，避免了类似错误的扩散。

功能与动机

根据 Issue #6068，当使用 use_fused_kernels=True 且 ulysses_sequence_parallel_size > 1 时，训练 reward 和 validation score 持续下降，而 SP=1 正常。分析发现 fused forward 在 SP 切片后执行 torch.roll(input_ids, shifts=-1)，导致本地切片的边界处标签错位。此 PR 修复该问题。

实现拆解

1. 模型适配器修改：在 dense_common、qwen3_vl、qwen2_vl、glm4v、qwen3_5 五个文件的 forward_with_torch_backend 和 forward_with_triton_backend 中添加 shift_labels 参数。标签计算逻辑改为优先使用 shift_labels（若不为 None），否则回退到本地 torch.roll。
2. FSDP engine 适配：在 verl/workers/engine/fsdp/transformer_impl.py 中，当 use_fused_kernels=True 且 use_remove_padding=True 时，将预计算的 output_args["input_ids_rmpad_rolled"] 作为 shift_labels 传递给 fused forward 的 extra_args。
3. 测试覆盖：新增 tests/models/test_fused_kernels_ulysses_sp_on_cpu.py，包含一个根因演示测试（slice-then-local-roll 与 global-roll-then-slice 对比）以及 10 个 adapter routing 测试（5 个模型 × 2 个后端），验证 shift_labels 被正确传递，以及缺失时回退行为。新增 tests/special_distributed/test_fused_kernels_ulysses_sp.py，在 2 GPU 上端到端验证 fused kernel + SP=2 与 SP=1 的 log prob 一致。

关键文件：

- tests/models/test_fused_kernels_ulysses_sp_on_cpu.py（模块 CPU 测试；类别 test；类型 test-coverage；符号 _global_then_slice, _slice_then_local_roll, test_local_roll_diverges_from_global_roll_under_sp, _FakeConfig）：新增 CPU 单元测

试，包含根因演示（slice-then-local-roll 与 global-roll-then-slice 对比）和 10 个适配器路由测试，验证 shift_labels 参数传递的正确性。

- tests/special_distributed/test_fused_kernels_ulysses_sp.py（模块 分布式测试；类别 test；类型 test-coverage；符号 _make_model, _broadcast_params, _init_dist, test_fused_kernels_log_probs_match_under_sp）：新增 2-GPU 分布式集成测试，端到端验证 fused kernel + SP=2 下 log prob 与 SP=1 一致。
- verl/models/transformers/dense_common.py（模块 通用适配器；类别 source；类型 data-contract；符号 forward_with_torch_backend, forward_with_triton_backend）：核心模型适配器，在两个 fused forward 函数中添加 shift_labels 参数并修改标签计算逻辑，是修复的主要代码位置。
- verl/models/transformers/qwen3_vl.py（模块 Qwen3-VL；类别 source；类型 data-contract；符号 forward_with_torch_backend, forward_with_triton_backend）：Qwen3-VL 模型的 fused forward 函数添加 shift_labels 参数，是 issue 报告中受影响的主要模型之一。
- verl/workers/engine/fsdp/transformer_impl.py（模块 FSDP 引擎；类别 source；类型 core-logic）：FSDP engine 将预计算的 input_ids_rmpad_rolled 通过 shift_labels 传递给 fused forward，是实现修复的引擎侧关键改动。

关键符号：forward_with_torch_backend, forward_with_triton_backend

关键源码片段

verl/models/transformers/dense_common.py

核心模型适配器，在两个 fused forward 函数中添加 shift_labels 参数并修改标签计算逻辑，是修复的主要代码位置。

```
def forward_with_torch_backend(
    self,
    input_ids: torch.LongTensor = None,
    attention_mask: Optional[torch.Tensor] = None,
    position_ids: Optional[torch.LongTensor] = None,
    past_key_values: Optional[Union["Cache", list[torch.FloatTensor]]] = None,
    inputs_embeds: Optional[torch.FloatTensor] = None,
    labels: Optional[torch.LongTensor] = None,
    use_cache: Optional[bool] = None,
    output_attentions: Optional[bool] = None,
    output_hidden_states: Optional[bool] = None,
    return_dict: Optional[bool] = None,
    cache_position: Optional[torch.LongTensor] = None,
    logits_to_keep: int | torch.Tensor = 0,
    temperature: float = 1.0,
    shift_labels: Optional[torch.LongTensor] = None, #
    新增参数：引擎预滚动的全局标签，用于避免本地 roll 的 SP 错误
    **loss_kwargs,
) -> tuple | CausalLMOutputForPPO:
    from verl.utils.experimental.torch_functional import FusedLinearForPPO
```

```
    outputs = forward_base_model(
        self,
        input_ids=input_ids,
        attention_mask=attention_mask,
        position_ids=position_ids,
        past_key_values=past_key_values,
        inputs_embeds=inputs_embeds,
        use_cache=use_cache,
        output_attentions=output_attentions,
        output_hidden_states=output_hidden_states,
        cache_position=cache_position,
    )

    hidden_states = outputs[0]

    if not return_dict:
        raise NotImplementedError("forward_with_torch_backend has to return_dict")

    # 标签计算：优先使用引擎传入的 shift_labels（已全局滚动并切片）
    # 这是修复 issue #6068 的关键：避免在已 SP 切片的 input_ids 上再次滚动
    if shift_labels is not None:
        rolled_labels = shift_labels
    elif labels is not None:
        rolled_labels = torch.roll(labels, shifts=-1, dims=-1)
    elif input_ids is not None:
        rolled_labels = torch.roll(input_ids, shifts=-1, dims=-1)
    else:
        raise RuntimeError("To use forward_with_torch_backend, either labels or input_ids must
        be provided.")

    fused_linear_for_ppo = FusedLinearForPPO()
    log_probs, entropy = fused_linear_for_ppo.forward(
        hidden_states=hidden_states,
        vocab_weights=self.lm_head.weight,
        input_ids=rolled_labels,
        temperature=temperature,
    )

    return CausalLMOutputForPPO(
        log_probs=log_probs,
        entropy=entropy,
        past_key_values=outputs.past_key_values,
        hidden_states=outputs.hidden_states,
        attentions=outputs.attentions,
    )
```

评论区精华

主要讨论来自审核者 wuxibin89 要求提交者格式化代码（按 CONTRIBUTING.md 中的 linting 规则）。提交者 shivam2199 按要求执行了 ruff 格式化，并申请 workflow 批准，最终获得批准。没有其他实质性的设计争议。

- 代码格式化要求 (style): 提交者 shivam2199 执行了 ruff 格式化，并通过了 CI。

风险与影响

- 风险：该修改向后兼容：shift_labels 默认 None，现有调用方（SP=1 或未使用 fused kernels）行为不变。新增测试覆盖了 SP=2 的端到端场景和单元测试，显著降低回归风险。需注意的潜在风险：fsdp engine 的改动仅适用于 rmpad 路径，非 rmpad 路径不受影响；Megatron engine 使用不同的 fused 路径，已确认不受影响。总体风险较低。
- 影响：影响所有使用 use_fused_kernels=True 且 ulysses_sequence_parallel_size > 1 的 FSDP 后端用户，修复了训练质量下降问题。对 SP=1、不使用 fused kernels 或使用 Megatron 引擎的用户无影响。团队需要确保自定义模型适配器也支持 shift_labels 参数（若使用 fused kernels），否则 engine 传递的 shift_labels 会被 **kwargs 忽略，但当前所有已适配模型均已修改。
- 风险标记：向后兼容，测试覆盖充分

关联脉络

- 暂无明显关联 PR