PR #6386 完整报告

verl-project/verl

[fsdp] fix: emit distillation outputs in use_remove_padding=False path

合并时间：2026-05-18 18:17

原文链接：http://prhub.com.cn/verl-project/verl/pull/6386

执行摘要

- 一句话：重新应用 no-rmpad 蒸馏修复并修复 CPU 测试
- 推荐动作：值得精读，特别是测试中通过 patch 隔离外部依赖（Triton 内核）的做法，这为编写跨平台的单元测试提供了可借鉴的模式。生产修复虽小但正确，体现了对两个分支对称性的重视。

功能与动机

6293 报告了 use_remove_padding=False 路径缺失 distillation top-k 处理，导致 distillation_losses 缺失并触发 KeyError。#6350 尝试修复但 #6360 因 CPU 测试失败（Triton CrossEntropyLoss 只能在 CUDA 上运行）而 revert。本 PR 重新应用生产修复并修正测试，确保两个 padding 分支的对称性。

实现拆解

1. 在 verl/workers/engine/fsdp/transformer_impl.py 的 prepare_model_outputs 方法中，在 no-rmpad 路径（else 分支末尾）增加了 distillation_use_topk 的处理逻辑，镜像 use_remove_padding=True 分支的做法：调用 logits_processor_func 得到输出，将每个 key 的 tensor 变换为嵌套 tensor 放入 model_output。
2. 新增测试文件 tests/workers/test_distillation_topk_symmetry_on_cpu.py，创建一个最小的 engine stub，用 object.__new__ 绕过初始化，并分别测试 use_remove_padding=True 和 False。测试通过 patch 拦截 logprobs_from_logits，返回一个形状正确的零张量，避免调用 flash-attn 的 Triton 内核。
3. 验证了蒸馏 keys 的传播正确性。

关键文件：

- verl/workers/engine/fsdp/transformer_impl.py（模块 FSDP 引擎；类别 source；类型 core-logic；符号 prepare_model_outputs）：核心生产代码修复，添加 no-rmpad 路径下的 distillation top-k 输出传播

- tests/workers/test_distillation_topk_symmetry_on_cpu.py（模块 蒸馏测试；类别 test；类型 test-coverage；符号 _make_engine_stub, _EngineCfg, _make_logits_processor, _proc）：新增回归测试，确保两个 padding 分支蒸馏输出一致性，并修复了 CPU 兼容性问题

关键符号：FSDPEngineWithLMHead.prepare_model_outputs, test_distillation_outputs_emitted_in_both_padding_modes

关键源码片段

verl/workers/engine/fsdp/transformer_impl.py

核心生产代码修复，添加 no-rmpad 路径下的 distillation top-k 输出传播

```
# 计算 log_probs 后
log_probs = logprobs_from_logits(logits=logits_rmpad, labels=input_ids_rmpad_rolled)

# 镜像 use_remove_padding=True 分支的处理（参见 verl#6293）
# 这里没有 Ulysses SP gather，因为本分支是 no-SP 路径
# （log_probs 也没有被 gather），且 pad_size 只在
# prepare_model_inputs 的 use_remove_padding=True 路径中填充到 output_args
if distillation_use_topk:
    outputs = logits_processor_func(student_logits=logits_rmpad.unsqueeze(0), data=micro_
    batch)
    for k, v in outputs.items():
        v = v.squeeze(0)
        assert v.shape == log_probs.shape, (
            f'log_probs shape: {log_probs.shape}, {k} shape: {v.shape}'
        )
        model_output[k] = torch.nested.nested_tensor_from_jagged(v, cu_seqlens)

# (bsz, j1), for each sample, length: [real_prompt_length + real_response_length]
log_probs = torch.nested.nested_tensor_from_jagged(log_probs, cu_seqlens)
```

tests/workers/test_distillation_topk_symmetry_on_cpu.py

新增回归测试，确保两个 padding 分支蒸馏输出一致性，并修复了 CPU 兼容性问题

```
@pytest.mark.parametrize('use_remove_padding', [True, False])
def test_distillation_outputs_emitted_in_both_padding_modes(use_remove_padding):
    '''验证蒸馏输出在两个 padding 分支都能正确传播。参见 verl#6293。'''
    bsz = 2
    seq_lengths_list = [3, 2]
    seq_lengths = torch.tensor(seq_lengths_list, dtype=torch.int64)
    total_nnz = int(seq_lengths.sum())
    cu_seqlens = torch.cat([torch.tensor([0]), seq_lengths.cumsum(0)]).to(torch.int64)
    # ... 准备 input_ids, output 等 ...

    # 设置 micro_batch 参数
    micro_batch = TensorDict({'input_ids': input_ids_nested}, batch_size=[])
    tu.assign_non_tensor(micro_batch,
```

```
                    use_remove_padding=use_remove_padding,
                    pad_mode=DatasetPadMode.NO_PADDING,
                    use_fused_kernels=False,
                    calculate_entropy=False,
                    calculate_sum_pi_squared=False,
                    distillation_use_topk=True,
                    max_response_length=max(seq_lengths_list))

    eng = _make_engine_stub()

    # 使用 patch 替换 logprobs_from_logits，避免在 CPU 上调用 Triton 内核
    with patch('verl.workers.engine.fsdp.transformer_impl.logprobs_from_logits',
               return_value=torch.zeros(total_nnz)):
        model_output = eng.prepare_model_outputs(
            output, output_args, micro_batch,
            logits_processor_func=_make_logits_processor(_DISTILLATION_KEYS),
            logits_processor_func_requires_grad=False,
            loss_function=None)

        # 断言蒸馏 keys 在 model_output 中存在且形状匹配
        expected_keys = {'distillation_losses', 'student_mass'}
        assert expected_keys.issubset(model_output.keys()), \
            f'Missing distillation keys: {expected_keys - set(model_output.keys())}'
        for k in expected_keys:
            v = model_output[k]
            assert isinstance(v, torch.Tensor), f'{k} should be tensor, got {type(v)}'
            assert v.dim() == 2, f'{k} dim should be 2, got {v.dim()}'
```

评论区精华

没有 review 评论，但作者在 PR body 中解释了 #6350 被 revert 的原因（CPU 测试失败），并说明了本 PR 的测试修复方案（patch logprobs_from_logits）。原作者 wuxibin89 批准了该 PR。

- CPU 测试环境下的 Triton 内核兼容性 (testing): 本 PR 通过 patch 在测试中替换 logprobs_from_logits 为返回零张量的存根，避免调用 Triton 内核，从而通过测试。

风险与影响

- 风险：核心逻辑变更是增添了一个条件分支（distillation_use_topk），如果使用该功能但未提供正确的 logits_processor_func，可能引发错误。不过该函数应在调用前配置好，风险可控。测试通过存根隔离了外部依赖，不会影响其他测试。没有性能或安全性风险。
- 影响：影响范围限于使用 FSDP 引擎且开启蒸馏训练的用户，特别是那些使用 use_remove_padding=False 配置的。这些用户之前会遇到 KeyError 错误，现在可以正常运行。影响程度较小，因为蒸馏功能还不是主线默认配置。
- 风险标记：核心路径变更，测试依赖注入

关联脉络

- PR #6350 [fsdp] fix: emit distillation outputs in use_remove_padding=False path: 原始修复 PR，本 PR 重新应用其生产变更
- PR #6360 Revert "[fsdp] fix: emit distillation outputs in use_remove_padding=False path (#6350)": 因 CPU 测试失败而 revert，本 PR 修复了测试
- PR #6293 [bug] use_remove_padding=False 路径缺失 distillation top-k 处理，导致 distillation_losses 缺失并触发 KeyError: 本 PR 修复的根因 issue