

PR #6350 完整报告

verl-project/verl

[fsdp] fix: emit distillation outputs in use_remove_padding=False path (#6293)

合并时间: 2026-05-14 20:55

原文链接: <http://prhub.com.cn/verl-project/verl/pull/6350>

执行摘要

- 一句话: 修复 no-rmpad 下 distillation top-k 输出缺失
- 推荐动作: 该 PR 值得精读, 原因有两点: 1) 展示了如何处理主干分支之间的对称性 bug, 以及对 review 反馈的快速响应; 2) 通过添加一个简洁的 CPU 单元测试有效地捕获了回归, 测试设计 (使用桩引擎、fake processor、两种 padding 输入构造) 可作为同类测试的参考。

功能与动机

Issue #6293 报告了在 `use_remove_padding=False` 配置下使用蒸馏 top-k 时, 由于 `prepare_model_outputs` 缺少处理块, 下游损失函数无法找到 `distillation_losses` 键而抛出 `KeyError`。PR body 明确指出这是 `False` 分支缺少对称处理导致的 bug。

实现拆解

1. 核心修复: 在 `verl/workers/engine/fsdp/transformer_impl.py` 的 `prepare_model_outputs` 方法的 `use_remove_padding=False / NO_PADDING` 路径中, 在计算完 `log_probs = logprobs_from_logits(...)` 后, 增加 `if distillation_use_topk:` 块, 调用 `logits_processor_func` 并将返回的字典逐 key 通过 `nested_tensor_from_jagged` 写入 `model_output`。
2. 去掉不必要的 SP 保护: 根据 review 讨论, 该路径实际上是 no-SP 路径, `pad_size` 不在 `output_args` 中, 因此删除了最初从 `True` 分支复制而来的 `if self.use_ulysses_sp:` 守卫, 直接进行嵌套张量包装。
3. 回归测试: 新增 `tests/workers/test_distillation_topk_symmetry_on_cpu.py`, 使用 `object.__new__` 创建桩引擎, 覆盖 `use_remove_padding=True` 和 `False` 两种模式, 断言所有蒸馏键均出现在 `model_output` 中, 且为嵌套张量。测试设计了两种不同的输入形状, 并利用 CPU 可执行的 fake logits_processor 避免 GPU 依赖。

关键文件:

- `verl/workers/engine/fsdp/transformer_impl.py` (模块引擎核心; 类别 source; 类型 core-logic; 符号 `prepare_model_outputs`): 核心修复文件, 在 `prepare_model_outputs` 方法的 `NO_PADDING` 分支中新增了蒸馏 top-k 处理, 与 `use_remove_padding=True` 分支对称。

- tests/workers/test_distillation_topk_symmetry_on_cpu.py (模块 测试覆盖; 类别 test; 类型 test-coverage; 符号 _make_engine_stub, _EngineCfg, _make_logits_processor, _proc) : 新增的 CPU 回归测试, 验证两种 padding 模式下蒸馏 top-k 输出都存在, 确保修复有效且防止未来回归。

关键符号: prepare_model_outputs, test_distillation_outputs_emitted_in_both_padding_modes, _make_engine_stub, _make_logits_processor

关键源码片段

verl/workers/engine/fsdp/transformer_impl.py

核心修复文件, 在 prepare_model_outputs 方法的 NO_PADDING 分支中新增了蒸馏 top-k 处理, 与 use_remove_padding=True 分支对称。

```
def prepare_model_outputs(self, output, output_args, micro_batch, logits_processor_func):
    # ... 前面的逻辑 ...
    else: # not using rmpad and no ulysses sp
        # ...
        if pad_mode == DatasetPadMode.NO_PADDING:
            cu_seqlens = input_ids.offsets()
            seq_lengths = cu_seqlens.diff()
            starts = torch.zeros_like(seq_lengths, dtype=torch.int64)
            logits = torch.nested.narrow(logits, 1, starts, seq_lengths, layout=torch.jagged)
            logits_rmpad = torch.cat([t for t in logits.unbind()])
            input_ids_rmpad_rolled = output_args["input_ids_rmpad_rolled"]
            log_probs = logprobs_from_logits(logits=logits_rmpad, labels=input_ids_rmpad_rolled)

            # Mirror the use_remove_padding=True branch (see verl#6293).
            # No Ulysses SP gather here: this branch is the no-SP path
            # (log_probs is also not gathered) and pad_size is only
            # populated in output_args along the use_remove_padding=True
            # path of prepare_model_inputs.
            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 of each sample: [real_prompt_length + real_
            response_length]
            log_probs = torch.nested.nested_tensor_from_jagged(log_probs, cu_seqlens)
            # ... 后续处理 ...
        # ...
    return model_output
```

评论区精华

gemini-code-assist[bot] 在 review 中指出：最初从 `use_remove_padding=True` 分支复制的 `if self.use_ulysses_sp:` 块会尝试访问 `output_args['pad_size']`，但该键只在 `True` 分支的 `prepare_model_inputs` 中填充，`False` 路径下会导致 `KeyError`；此外，`False` 路径本身不执行 Ulysses SP gather，因此内部 gather 会导致蒸馏 key 与 `log_probs` 张量形状不匹配。abinggo 确认了问题，并提交了第二次 commit (2826ee19)，完全删除 `if self.use_ulysses_sp:` 保护，直接使用 `nested_tensor_from_jagged` 包装蒸馏输出，与同路径下 `log_probs` 的处理方式一致。

- no-rmpad 路径下 `pad_size` `KeyError` 及 SP 不必要 (correctness): abinggo 确认问题后，提交了 commit 2826ee19，完全删除 `if self.use_ulysses_sp:` 保护，直接使用 `nested_tensor_from_jagged` 包装蒸馏输出，与 `log_probs` 处理方式一致。

风险与影响

- 风险：该修复仅影响 `use_remove_padding=False + distillation_use_topk=True` 这一特定组合路径，且改动范围局限在 `prepare_model_outputs` 的一个分支内，有完整的 CPU 单元测试覆盖。风险极低，不会影响默认的 `use_remove_padding=True` 路径或其他功能。需要注意的是，测试中依赖的 `logprobs_from_logits` 在 CPU 上可能触发 Triton 内核（如 `flash-attn`），已在后续 PR #6386 中专门规避，但当前 PR 的测试在 CI CPU 任务中曾因 Triton 失败，最终由后续更新解决。
- 影响：该 fix 直接修复了使用 `use_remove_padding=False` 配置进行蒸馏训练时的崩溃问题。所有使用 `NO_PADDING` 模式 + `distillation_use_topk=True` 的用户均受影响，此前训练无法进行。修复后，该配置可正常运行，输出行为与 `use_remove_padding=True` 分支一致。对未使用蒸馏 top-k 或使用默认 padding 模式的用户无影响。
- 风险标记：测试环境依赖，特定配置路径

关联脉络

- 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`: 关联的 issue，报告了该 bug，是整个 PR 的动机来源。
- PR #6360 Revert "[fsdp] fix: emit distillation outputs in `use_remove_padding=False` path (#6293)": 回退了本 PR，因 CI 中 CPU 测试遇到 Triton 错误，作为快速恢复措施。
- PR #6386 [fsdp] fix: emit distillation outputs in `use_remove_padding=False` path (#6293) with CPU-safe test: 重新应用了本 PR 的修复，并调整测试避免 GPU 依赖，是最终干净的版本。