

PR #6290 完整报告

verl-project/verl

[fsdp] fix: lenient resolution of `_no_split_modules` in `get_fsdp_wrap_policy`

合并时间: 2026-05-14 08:39

原文链接: <http://prhub.com.cn/verl-project/verl/pull/6290>

执行摘要

- 一句话: 放宽 FSDP wrap 层名解析, 支持部分缺失
- 推荐动作: 此 PR 修复了一个真实阻碍训练的问题, 代码简洁, 测试覆盖良好, 推荐集成。
但建议在新 PR 中跟进 reviewer 提出的字符串类型归一化建议, 以确保从配置文件传入字符串时的鲁棒性。

功能与动机

HuggingFace transformers 有时在 `_no_split_modules` 中同时包含有效和向前兼容的类名 (如 Qwen3.5 列出 `Qwen3_5DecoderLayer` 和 `Qwen3_5TextDecoderLayer`, 后者尚未存在)。原实现遇到第一个缺失就抛出异常, 导致 FSDP 训练完全中断。本 PR 使解析过程宽容, 只要至少一个类名能解析就继续, 并记录警告。

实现拆解

1. 修改核心循环: 在 `verl/utils/fsdp_utils.py` 的 `get_fsdp_wrap_policy` 中, 将之前对每个层名解析失败直接 `raise` 改为收集到 `missing_classes` 列表, 并继续解析后续名称。
2. 后置检查与反馈: 循环结束后, 如果 `transformer_cls_to_wrap` 为空集合, 说明没有一个名称能解析, 则抛出 `Exception` 并提示信息中包含完整的 `fsdp_transformer_layer_cls_to_wrap` 列表。否则, 如果存在缺失名称, 则使用 `logger.warning` 输出警告, 明确列出跳过的名称和实际 wrap 的类名。
3. 添加日志基础设施: 在文件顶部 `import logging` 并创建模块级 `logger = logging.getLogger(__name__)`。
4. 新增单元测试: 在 `tests/utils/test_fsdp_wrap_policy_on_cpu.py` 中创建三个测试用例:
 - `test_wrap_policy_skips_missing_layer_class_names`: 部分解析成功场景, 验证警告信息。
 - `test_wrap_policy_raises_when_no_layer_classes_resolve`: 全部解析失败场景, 验证异常。
 - `test_wrap_policy_explicit_config_overrides_no_split_modules`: 显式配置覆盖场景, 验证覆盖生效。

关键文件:

- verl/utils/fsdp_utils.py (模块 FSDP 工具; 类别 source; 类型 core-logic; 符号 get_fsdp_wrap_policy) : 核心修改: 放宽了 FSDP wrap policy 中层名称解析的严格性, 允许部分失败并发出警告。
- tests/utils/test_fsdp_wrap_policy_on_cpu.py (模块 FSDP 测试; 类别 test; 类型 test-coverage; 符号 _RealLayer, _PartiallyResolvableModel, _AllUnresolvableModel, test_wrap_policy_skips_missing_layer_class_names) : 新增 CPU 单元测试, 验证 lenient resolution 的正确性, 三种场景全覆盖。

关键符号: get_fsdp_wrap_policy

关键源码片段

verl/utils/fsdp_utils.py

核心修改: 放宽了 FSDP wrap policy 中层名称解析的严格性, 允许部分失败并发出警告。

```
elif fsdp_transformer_layer_cls_to_wrap is not None:
    transformer_cls_to_wrap = set()
    missing_classes = [] # 收集无法解析的类名
    for layer_class in fsdp_transformer_layer_cls_to_wrap:
        transformer_cls = get_module_class_from_name(module, layer_class)
        if transformer_cls is None:
            # HuggingFace transformers 有时在 `_no_split_modules` 中同时包含
            # 有效的类名和尚未导出的向前兼容名称 (如 Qwen3.5 的
            # `Qwen3_5TextDecoderLayer`)。逐个跳过未知名称,
            # 而不是立即抛出异常, 确保至少有一个名称能解析即可构建策略。
            missing_classes.append(layer_class)
        else:
            transformer_cls_to_wrap.add(transformer_cls)

    if not transformer_cls_to_wrap:
        raise Exception(
            'Could not find any of the transformer layer classes to wrap in the model: '
            f'{list(fsdp_transformer_layer_cls_to_wrap)}'
        )
    if missing_classes:
        logger.warning(
            'FSDP wrap policy: skipped missing layer class names %s, wrapping %s',
            missing_classes,
            sorted(c.__name__ for c in transformer_cls_to_wrap),
        )

    transformer_policy = functools.partial(
        transformer_auto_wrap_policy,
        transformer_layer_cls=transformer_cls_to_wrap,
    )
    policies.append(transformer_policy)
```

评论区精华

Reviewer [gemini-code-assist\[bot\]](#) 提出: `fsdp_transformer_layer_cls_to_wrap` 可能从 YAML/OmegaConf 配置中作为字符串传入, 直接迭代会导致逐字符解析错误, 建议像 `apply_fsd2` 那样进行类型归一化。该评论在合并时未解决, 后续需关注。

- 字符串类型归一化缺失风险 (correctness): 该评论未在 PR 中得到解决, PR 已合并。后续可能需要单独 Issue 或 PR 跟进。

风险与影响

- 风险:
 - 字符串类型未处理风险: 若配置传入字符串, 循环将逐字符迭代, 几乎必然失败。建议跟进归一化处理。
 - 宽松解析隐蔽错误风险: 部分无效名称被跳过可能掩盖真实配置错误, 但警告日志提供了可观测性。
 - 回归风险低: 修改仅针对解析失败路径, 对可成功解析的场景无影响。
- 影响:
 - 用户影响: 使用 HuggingFace 模型且依赖 `_no_split_modules` 自动构建 FSDP wrap policy 的用户受益, 特别是 Qwen3.5 系列模型现在可以顺利进行 FSDP 训练。没有用户受影响 (因为之前总是失败)。对于其他模型, 行为基本不变, 只是启动时若存在无效名称会多一条警告日志。
 - 系统影响: 训练脚本无需修改, 向后兼容。FSDP 工具函数行为更健壮。
 - 团队影响: 减少因 HF 模型更新导致的兼容性问题工单, 降低维护成本。
 - 风险标记: 字符串类型未防御, 宽松解析隐蔽错误

关联脉络

- PR #6289 [fsdp] FSDP wrap policy raises on partial-resolution `_no_split_modules` (Qwen3.5 / Qwen3-Next case): 关联 Issue, 直接描述本 PR 解决的问题。
- PR #6317 [fsdp] fix: FSDP2 silently drops `fsdp_config.forward_prefetch`: 同一文件 (`fsdp_utils.py`) 的近期 bugfix, 体现 FSDP 配置修复的持续演进。