

PR #2213 完整报告

radixark/miles

fix(fsdp): apply the GDN packing patch to dense qwen3_5 and fix patch

合并时间: 2026-08-06 05:01

原文链接: <http://prhub.com.cn/radixark/miles/pull/2213>

执行摘要

- 一句话: 修复 dense Qwen3.5 的 GDN packing 状态泄漏与边界残留
- 推荐动作: 值得精读。两个设计点值得借鉴: 1) 以 decoder wrapper 为 producer、GDN wrapper 为 consumer 的边界 mailbox 模式, 以及“无条件写入 + None 清理”同时解决泄漏与陈旧状态; 2) “packing fix applied 日志由其他架构触发”导致的静默跳过陷阱——日志不足以证明覆盖范围, 枚举架构时应反向断言每个目标都被 patch。若团队继续维护 FSDP 后端, 建议为 packing patch 增加 fast 级单元测试。

功能与动机

PR body 明确指出两个动机: 一是 dense Qwen3.5 的建模代码位于独立的 transformers.models.qwen3_5 模块 (与 qwen3_5_moe/qwen3_next 分属不同模块), 原 patch 循环只遍历后两者, dense 架构从未被打上 packing patch, 但 "packing fix applied" 日志仍会被 moe/next 类触发, 形成“日志显示已修复、实际未覆盖”的静默跳过; 二是边界属性 _gdn_cu_seqlens/_gdn_seq_idx 是 decoder wrapper (生产者) 与 GDN wrapper (消费者) 之间的 mailbox, 原实现只在 packed forward 时写入, 当单个样本长度恰为 pad 整数倍时会出现无 pad 伪文档的单文档行, 沿用上一行旧边界会导致 conv kernel 崩溃 (seq_idx must have shape (batch_size, seqlen)) 或静默错位。

实现拆解

1. 扩展 GDN patch 覆盖范围: 在 miles/backends/experimental/fsdp_utils/models/qwen3_5.py 的 apply_gateddeltanet_packing_patch() 中, 模块循环由 ("qwen3_5_moe", "qwen3_next") 扩展为 ("qwen3_5", "qwen3_5_moe", "qwen3_next"), 使 dense Qwen3.5 的 GatedDeltaNet 与 DecoderLayer 类也被 _patch_gdn_forward 和 _patch_decoder_forward 包装, 修复其 linear-attention 层在 THD packing 下的状态泄漏。
2. 边界属性无条件写入: _patch_decoder_forward 内的 forward 包装移除 if ctx is not None 短路, 改为每次前向都遍历 self.modules(), 对每个 GDN 层写入 ctx.cu_seqlens/ctx.seq_idx (packed 行) 或 None (非 packed 行)。None 写入让 GDN wrapper 走到干净路径, 等价于单文档前向, 避免陈旧几何信息残留。
3. 重命名对齐: 按 reviewer 建议将 models/qwen3_5_moe.py、adaptations/specs/qwen3_5_moe.py 重命名为 qwen3_5.py, 并更新 adaptations/specs/__init__.py 的导入 (qwen3_5_moe → qwen3_5), 保持 "imports

trigger registration” 的注册链路与 spec docstring 中的路径引用同步。

4. 验证与配套：未新增自动化测试（PR body 注明按 review 移除，行为由 8xH200 live GRPO wandb 对比曲线和 eager 模式隔离测试覆盖）；无新增 launch flags，无需更新 CLI reference 文档。

关键文件：

- miles/backends/experimental/fsdp_utils/models/qwen3_5.py（模块 FSDP 适配；类别 source；类型 core-logic；符号 _patch_decoder_forward, apply_gateddeltanet_packing_patch）：核心修复文件：把 dense qwen3_5 模块纳入 GDN packing patch 的模块循环，并将 decoder wrapper 的边界 mailbox 写入改为无条件执行（非 packed 写 None），解决状态泄漏与陈旧边界残留。
- miles/backends/experimental/fsdp_utils/adaptations/specs/qwen3_5.py（模块 适配注册；类别 source；类型 rename-or-move；符号 _apply, _applies）：架构适配 spec 的注册入口：_apply() 的导入路径与 docstring 同步从 qwen3_5_moe 更新为 qwen3_5，是 packing patch 注册链路的必要配套。
- miles/backends/experimental/fsdp_utils/adaptations/specs/__init__.py（模块 适配注册；类别 source；类型 dependency-wiring）：注册包导入从 qwen3_5_moe 改为 qwen3_5，维持 "imports trigger registration" 机制，保证重命名后各架构 spec 仍被注册。

关键符号：_patch_decoder_forward, apply_gateddeltanet_packing_patch

关键源码片段

miles/backends/experimental/fsdp_utils/models/qwen3_5.py

核心修复文件：把 dense qwen3_5 模块纳入 GDN packing patch 的模块循环，并将 decoder wrapper 的边界 mailbox 写入改为无条件执行（非 packed 写 None），解决状态泄漏与陈旧边界残留。

```
# miles/backends/experimental/fsdp_utils/models/qwen3_5.py（由 qwen3_5_moe.py 重命名而来）
```

```
def _patch_decoder_forward(dl_cls, gdn_cls):
    orig = dl_cls.forward
    if getattr(orig, "_gdn_packing", False):
        return # 幂等：同一 DecoderLayer 类只 patch 一次

    @functools.wraps(orig)
    def forward(self, *args, **kwargs):
        # 每次前向无条件刷新边界 mailbox:
        # 非 packed 行（如单样本长度恰为 pad 倍数）必须清为 None,
        # 否则上一行残留的 cu_seqlens/seq_idx 会导致 conv kernel 崩溃或静默错位。
        ctx = packed_seq_context(kwargs.get("position_ids"))
        for module in self.modules():
            if isinstance(module, gdn_cls):
                module._gdn_cu_seqlens = ctx.cu_seqlens if ctx is not None else None
                module._gdn_seq_idx = ctx.seq_idx if ctx is not None else None
        return orig(self, *args, **kwargs)
```

```
forward._gdn_packing = True
dl_cls.forward = forward
```

```
def apply_gateddeltanet_packing_patch():
    """Patch 每个含 GatedDeltaNet 的混合架构（幂等）。返回是否 patch 了任何类。"""
    patched = False
    # 覆盖 dense qwen3_5（独立 transformers 模块）、qwen3_5_moe 与 qwen3_next;
    # 若漏掉 dense 模块，“packing fix applied”日志仍会被 moe/next 触发，形成静默跳过。
    for mod_name in ("qwen3_5", "qwen3_5_moe", "qwen3_next"):
        try:
            mod = __import__(f"transformers.models.{mod_name}.modeling_{mod_name}", fromlist=
                ["x"])
        except Exception:
            continue # 当前环境未安装该架构时跳过
        gdn_cls = _find_class(mod, "GatedDeltaNet")
        dl_cls = _find_class(mod, "DecoderLayer")
        if gdn_cls is None or dl_cls is None:
            continue
        _patch_gdn_forward(gdn_cls)
        _patch_decoder_forward(dl_cls, gdn_cls)
        patched = True

    if patched:
        logger.info(
            "[fsdp] GatedDeltaNet packing fix applied: cu_seqlens/seq_idx reset the "
            "linear-attn recurrence and causal-conv state per packed document"
        )
    return patched
```

评论区精华

唯一的 review 讨论来自 Zhichenzzz 的评论“rename the file qwen3_5_moe to qwen?”，作者在 issue 评论中回复“Renamed qwen3_5_moe -> qwen3_5 for all related files”。这是命名 / 维护性建议（dense 与 MoE 架构不应共用一个带 `_moe` 后缀的文件），已在第二个 commit 中通过重命名 `models/qwen3_5_moe.py` 与 `adaptations/specs/qwen3_5_moe.py` 为 `qwen3_5.py` 并同步更新导入解决，Zhichenzzz 最终 APPROVED。另据 PR body 自述，原计划新增的测试按 review 移除，改为以实测数据作为行为证据。

- qwen3_5_moe 文件是否应重命名为 qwen3_5 (design): 作者按建议将 `models/qwen3_5_moe.py` 与 `specs/qwen3_5_moe.py` 统一重命名为 `qwen3_5.py`，并同步更新 `specs/init.py` 导入与 docstring；Zhichenzzz 最终 APPROVED。

风险与影响

- 风险：
 1. 正确性依赖：无条件写入依赖 `packed_seq_context` 对 `position_ids` 的解析。若未来存在不经过 `position_ids` 触发的 packing 路径（如直接传入 `cu_seqlens`），可能漏判；不

过当前对 `ctx is None` 走清理路径，比旧实现更安全。

2. 性能：每次 `decoder forward` 都多一次 `self.modules()` 遍历（原 `packed` 行同样遍历，非 `packed` 行短路）。这是 $O(\text{层数})$ 的 Python 级扫描，对训练吞吐影响可忽略，但超长序列下需留意。
3. 兼容性：文件重命名会影响所有引用旧模块路径的代码；仓库内 `import` 已同步，但外部脚本或未合并分支若直接 `import miles.backends.experimental.fsdps_utils.models.qwen3_5_moe` 会得到 `ImportError`。
4. 缺少自动化测试：无 `fast/e2e` 测试断言 `_patch_decoder_forward` 在 `packed/non-packed` 两种输入下的属性写入行为，回归保障依赖人工实验；建议补充针对 `apply_gateddeltanet_packing_patch` 覆盖枚举的断言测试。
 - 影响：影响用户：使用 FSDP 后端做 Qwen3.5 dense RL 训练的任务将获得正确的 per-document 状态重置，train/rollout logprob 差异回到约 $6e-3$ 的 kernel 噪声水平，并消除 conv kernel 偶发崩溃。影响系统：改动集中在 `miles/backends/experimental/fsdp_utils`（实验性 FSDP 后端）的 THD packing 路径，不影响 Megatron 主路径，性能开销可忽略。影响团队：文件命名从 `qwen3_5_moe` 收窄为 `qwen3_5`，与 `transformers` 建模模块名对齐；模块循环显式覆盖三个架构，避免“日志发出但实际未生效”的静默跳过陷阱再次发生。
 - 风险标记：缺少测试覆盖，文件重命名，实验性后端，训练正确性

关联脉络

- PR #2031 feat(fsdp): add hybrid sharding: 同一 FSDP 实验后端的能力演进主线，本 PR 是对该后端 Qwen3.5 架构支持的后续正确性修复。
- PR #2045 codeowners: cover miles/backends/experimental/fsdp_utils: 该目录已被纳入 CODEOWNERS 责任区，本 PR 的改动正落在此维护分区内。