PR #47640 完整报告

vllm-project/vllm

[Bugfix][LoRA] Guard None group members in expand_packed_lora (partial LoRA on Qwen3.5/3.6 GatedDeltaNet)

合并时间：2026-08-19 21:24

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/47640

执行摘要

- 一句话：修复 expand_packed_lora 对 None 组员的崩溃
- 推荐动作：值得精读，因为它是理解 vLLM LoRA 打包层 None 容错模式的精简案例，展示了如何在融合层中通过占位符保持切片对齐。关注点：expand_packed_lora 与 slice_lora_b、set_lora 的容错一致性；以及缺少自动化测试的隐患。对于 LoRA 开发者，这个模式可以复用到其他打包投影组。

功能与动机

该 PR 解决 Issue #47639 报告的崩溃：当 LoRA 适配器只针对 GatedDeltaNet 打包投影组的部分成员（如 in_proj_qkv 而不含 in_proj_z）时，expand_packed_lora 解引用 b_i.shape[0] 会在 None 上抛出 AttributeError。PR body 明确写道：'This is a regression introduced by #37912, which unified the GDN input projections into a fused packed layer and added expand_packed_lora — but the new helper didn't carry over the None-tolerance that its sibling slice_lora_b and the set_lora stacking loop already have.' 在 v0.20.0 中，部分组通过 create_in_proj_qkvz 构建为独立模块，可以正常工作；从 v0.21.0 起崩溃并持续到 v0.24.0 及当前 main。

实现拆解

1. 定位问题：在 vllm/lora/layers/column_parallel_linear.py 的 MergedColumnParallelLinearWithLoRA.expand_packed_lora 中，遍历 lora_a/lora_b 时对每个 b_i 直接调用 b_i.shape[0]，当 b_i 为 None（未适配的组员）时抛出 AttributeError。
2. 放宽类型签名：将 lora_a、lora_b 及返回值的类型从 list[torch.Tensor] 改为 list[torch.Tensor | None]，与同文件中的 slice_lora_b 和 set_lora 的 None 容错保持一致。
3. 增加 None 分支：在循环开头判断 b_i is None 时，推断其覆盖的切片数为 self.n_slices - start_idx（即剩余全部切片），为这些切片追加 None 占位符，然后 start_idx 累加并 continue。这保证了后续组的位置对齐，未适配切片保留基础权重。
4. 注释明确布局假设：代码注释说明该推断依赖 in_proj_qkvz 组唯一的布局——多切片成员 in_proj_qkv 在前、可选成员 in_proj_z 在后，因此 None 成员恰好覆盖剩余切片。
5. 测试与验证：作者最初在 PR 中包含 tests/lora/test_expand_packed_lora.py 回归测试，但在提交 ca0f6ddf 中移除，使 PR 仅含代码改动；作者在 1×H200、vLLM 0.24.0、

torch 2.11.0+cu130 上用真实 Qwen3.6-27B LoRA 适配器（目标 q_proj,k_proj,v_proj,in_proj_qkv）验证了修复前后行为。最终 PR 未包含自动化测试。

关键文件：

- vllm/lora/layers/column_parallel_linear.py（模块 LoRA 层；类别 source；类型 core-logic；符号 expand_packed_lora）：唯一变更文件，expand_packed_lora 是崩溃的直接位置；修复通过新增 None 分支恢复部分 LoRA 组支持。

关键符号：expand_packed_lora

关键源码片段

vllm/lora/layers/column_parallel_linear.py

唯一变更文件，expand_packed_lora 是崩溃的直接位置；修复通过新增 None 分支恢复部分 LoRA 组支持。

```
# 整理自 vllm/lora/layers/column_parallel_linear.py 的 expand_packed_lora
def expand_packed_lora(
    self,
    lora_a: list[torch.Tensor | None],
    lora_b: list[torch.Tensor | None],
) -> tuple[list[torch.Tensor | None], list[torch.Tensor | None]]:
    """
    Expand packed adapter groups when they don't match n_slices.
    E.g. in_proj_qkv (covers Q+K+V) + in_proj_z.

    A None group member means that member was not adapted; the slice(s)
    it covers are emitted as None placeholders so subsequent groups stay
    aligned and those slices are left at base weights.
    """
    expanded_a: list[torch.Tensor | None] = []
    expanded_b: list[torch.Tensor | None] = []
    start_idx = 0
    for a_i, b_i in zip(lora_a, lora_b):
        if b_i is None:
            # 未适配的组员：张量缺失导致无法读取行数，
            # 因此把覆盖范围推断为剩余的全部切片。
            # 该推断对唯一走此路径的融合 GDN in_proj_qkvz 组是精确的，
            # 因为多切片成员 in_proj_qkv（Q+K+V）在前，
            # 可选成员 in_proj_z 在后。
            covered = self.n_slices - start_idx
            for _ in range(covered):
                expanded_a.append(None)
                expanded_b.append(None)
            start_idx += covered
            continue
        # 确定这个 b_i 覆盖哪些输出切片。
        b_rows, cu_rows, covered = b_i.shape[0], 0, 0
        for i in range(start_idx, self.n_slices):
```

```
            cu_rows += self.output_sizes[i]
            if cu_rows == b_rows:
                covered = i - start_idx + 1
                break
        else:
            raise ValueError(
                f"Cannot determine how to split lora_b with {b_rows} rows "
                f"into {self.n_slices} slices with output sizes "
                f"{self.output_sizes} starting from index {start_idx}."
            )
        # 将 b_i 切成逐切片张量，并为每个切片复制 a_i。
        start = 0
        for j in range(covered):
            size = self.output_sizes[start_idx + j]
            expanded_b.append(b_i[start : start + size, :])
            expanded_a.append(a_i)
            start += size
        start_idx += covered
    return expanded_a, expanded_b
```

评论区精华

PR 本身没有实质性的 review 评论，维护者 linitra24 和 jeejeelee 直接批准。讨论主要来自 Issue #47639 评论：

- 作者多次更新版本未修复情况：> "Update：not fixed in vllm v0.25.0，pr still relevant"、> "still not fixed in vllm0.27.1"。
- 作者主动移除回归测试：> "I removed the test from the pr, may not be necessary, only 1 file (though a very important one) change."
- CI 失败后作者判断 "i think unrelated?"，经 /ci retry 与再次 /ci run 后 Buildkite CI #84597 通过。
- 回归测试移除 (testing): 测试被移除，PR 仅含代码改动；维护者批准时未要求补充测试。
- Bug 在多个版本未修复 (question): 维护者最终批准并合并，修复随后续版本发布。
- CI 失败与重试 (other): 重试后通过，未发现与改动相关的 CI 问题。

风险与影响

- 风险：技术风险集中在以下三点：
  1. 布局顺序假设：covered = self.n_slices - start_idx 的推断依赖 in_proj_qkvz 组内 in_proj_qkv 在前、in_proj_z 在后的固定顺序。若未来新增其他打包组或调整成员顺序，且存在 None 成员，该推断可能错误地覆盖切片。代码注释已说明这一假设，但缺少运行时校验。
  2. 缺少自动化测试：作者移除了回归测试，CI 无法覆盖部分 LoRA 加载场景，未来改动容易再次引入同类回归。手动验证仅在特定硬件与模型组合（H200、Qwen3.6-27B）上完成。

3. 非 None 路径行为未变：正常路径的逻辑与错误提示保持不变，但类型签名放宽后，调用方传入 None 的行为从崩溃变为静默跳过，可能掩盖调用方构造 lora_b 时的其他问题。总体回归风险较低，因为改动仅新增分支，不影响原有非 None 流程。

- 影响：影响范围集中在 LoRA 与 Qwen3.5/3.6 GatedDeltaNet 的组合场景：此前加载仅适配 in_proj_qkv（不含 in_proj_z）的适配器会直接崩溃，修复后恢复 v0.20.0 的部分组支持。对系统性能无影响（仅多一次 b_i is None 判断）。对团队而言，这是对 #37912 重构的兼容性修补，维护者需注意其依赖的布局假设，并考虑未来补充回归测试。
- 风险标记：缺少自动化测试覆盖，依赖打包组布局顺序假设

关联脉络

- PR #48850 [Bugfix][LoRA] Add embedding_modules for Qwen3.5 CausalLM: 同为 Qwen3.5 LoRA 支持层面的边界修复，涉及模型层结构假设的容错；说明 Qwen 系列模型的 LoRA 适配常需补充模型注册或层配置细节。