

PR #34446 完整报告

sgl-project/sglang

[rotary] Fix the fused Qwen3.5 RoPE kernel discarding mrope height and width

合并时间: 2026-08-30 15:37

原文链接: <http://prhub.com.cn/sgl-project/sglang/pull/34446>

执行摘要

- 一句话: 修复 fused RoPE 丢弃 mrope 高度宽度坐标
- 推荐动作: 值得精读, 尤其是 kernel 层与 rotary 层配合的写法。关注 4 个设计点:
 1. 用 MROPE: `tl.constexpr` 区分 1-D 与 mrope 特化, 既修复 bug 又不损失 1-D 性能 (两个 benchmark 数据佐证)。
 2. 行距取 `positions.stride(0)` 而不是 3, 兼容 CUDA-graph decode 从宽 buffer 切出的子行。
 3. `_build_axis_map` 将三种布局收敛为统一契约, 并用 `_legacy_axis_map` 隔离新旧 kernel 的输入差异。
 4. 测试特意使用三轴互不相同的位置 (`t != h != w`) 来捕捉“只取 row 0”类 bug, 并复现 CUDA-graph 的 buffer 切片场景。

功能与动机

PR body 明确指出根因: `fused_qk_gemma_rmsnorm_rope_gate` 对每个 token 只加载一个位置, 而多模态 Qwen3.5 传入的 mrope positions 是 `[3, T]` (每行一个轴), 偏移只落在 row 0, 图像 token 全部按时间轴位置旋转, height/width 被丢弃。该路径不受 flag 保护, `_is_cuda and attn_output_gate` (默认 True) 下 CUDA 上所有带图请求都会命中; 文本 token 三行相同所以文本基准无法暴露问题。

实现拆解

修复分四层推进:

1. Triton kernel 层 (`python/sglang/kernels/ops/attention/fused_qk_rmsnorm_rope_gate.py`) - `_fused_qk_rmsnorm_rope_gate_kernel` 新增 `mrope_axis_map_ptr`、`stride_pos_axis` 与 MROPE: `tl.constexpr` 三个参数; MROPE 分支下, 每个 rotary lane 先按 `mrope_axis_map_ptr + rot_offs` 读出所属轴, 再以 `axis * stride_pos_axis + token` 定位 position 并加载。- 行距由 `positions.stride(0)` 提供而不是常数 3, 因为 CUDA-graph decode 会以 `mrope_positions[:, :num_tokens]` 形式切出宽 buffer 的子行, 行 stride 仍是 buffer 的行距。- MROPE 是编译期常量, 1-D 路径保留独立特化, 不引入额外 launch、同步或分配。- `fused_qk_gemma_rmsnorm_rope_gate` Python 入口新增 `mrope_axis_map` 参数, 并做 4 组防御断言: `positions.dim() in (1, 2)`、mrope 与 map 必须成对出现、`[3, T]` 在 T 维连续、map 长度等于 `rotary_dim // 2`。

2. 旋转编码层 (python/sglang/srt/layers/rotary_embedding/mrope.py) - 把原来只有 `mrope_interleaved_glm` 分支构造 `axis_map` 的逻辑, 泛化成独立方法 `_build_axis_map`, 覆盖三种布局: GLM round-robin、Qwen 系 interleaved、contiguous section split; 返回的 `axis_map` 统一注册为 buffer。- 新增 `_legacy_axis_map` property: 只有 GLM 下返回 map, `forward_triton` 与 `forward_xpu` 改用它, 保证 `triton_mrope_fused` 和树外 `sgl_kernel.multimodal_rotary_embedding` 收到的输入与之前完全一致。- `Ernie4_5_VLRotaryEmbedding` 覆写 `_build_axis_map` 返回 `None`, 因为 Ernie 按 h、w、t 顺序读 `mrope_section`, 与共享 builder 的 t、h、w 假设冲突, 显式退出避免错位。
3. 模型接线层 (python/sglang/srt/models/qwen3_5.py) - `Qwen3_5AttentionDecoderLayer.forward_prepare_cuda_fused` 在调用 fused kernel 时按 `positions.dim() == 2` 条件传入 `self.rotary_emb.axis_map`, 一行改动使多模态路径真正吃到 map。
4. 测试与验证配套 - 新增 `test_fused_qk_rmsnorm_rope_gate.py`: 对 1-D positions 用 `gemma_rmsnorm + neox_rope` 参考实现对比; 对 mrope 用 `MRotaryEmbedding.forward_native` 做参考, 分别覆盖 interleaved/contiguous 的 [11, 11, 10] 和 [24, 20, 20], 并把 positions 从 4 倍宽 buffer 切片以复现 CUDA-graph 行 stride; 另测 1-D 与 map 不配套的非法组合必须抛 `AssertionError`。- 新增 `test_mrope_axis_map.py`: 用 `select_by_axis` 逐一与 `apply_interleaved_rope`、section split 对比; pinned GLM 的 round-robin 顺序 (其消费 kernel 在树外); 验证非 GLM 的 `_legacy_axis_map` 为 `None`、Ernie 的 `axis_map` 为 `None`。- PR body 给出 B200 上的精度与性能数据: 修复前 h 轴误差 5.891、w 轴 19.875 (bf16 下限 0.0625), 修复后三轴全部到下限; 端到端图片问答中, 修复前三个词的框 x1 都固定在 800, 修复后与 unfused 参考路径差 8 px 以内; mrope 特化成本在 3072 tokens 时为 -0.08%、decode 32 tokens 时为 +0.11%。

关键文件:

- python/sglang/srt/layers/rotary_embedding/mrope.py (模块 旋转编码; 类别 source; 类型 core-logic; 符号 `_build_axis_map`, `_legacy_axis_map`): `axis map` 统一构造是本修复的数据契约核心: 把 GLM-only 的 map 构造泛化到三种布局, 并提供 `_legacy_axis_map` 隔离新旧 kernel 的输入差异。
- python/sglang/kernels/ops/attention/fused_qk_rmsnorm_rope_gate.py (模块 融合内核; 类别 source; 类型 core-logic; 符号 `fused_qk_gemma_rmsnorm_rope_gate`, `_fused_qk_rmsnorm_rope_gate_kernel`): bug 的实际发生地; kernel 按 `axis map` 加载 mrope 位置是修复主战场, 且入口新增 mrope 契约断言。
- python/sglang/srt/models/qwen3_5.py (模块 模型接线; 类别 source; 类型 data-contract; 符号 `forward_prepare_cuda_fused`): 一行接线决定 fused 路径何时把 `axis_map` 传给 kernel, 是修复实际生效的模型侧入口。
- test/registered/kernels/ops/attention/test_fused_qk_rmsnorm_rope_gate.py (模块 融合内核测试; 类别 test; 类型 test-coverage; 符号 `TestFusedQKRMSNormRoPEGate`, `gemma_rmsnorm`, `neox_rope`, `build_mrope`): 新增的 kernel 级精度测试: 覆盖 1-D 回归、mrope 与 `forward_native` 对齐、非法位置 -map 组合拒绝, 以及 CUDA-graph 行 stride 复现。

- test/registered/rotary/test_mrope_axis_map.py (模块 旋转编码测试; 类别 test; 类型 test-coverage; 符号 TestMRopeAxisMap, build_mrope, select_by_axis) : 验证三种布局的 axis map 与消费方一致, pinned GLM round-robin 顺序, 并确认 Ernie opt-out 与 legacy map 契约。

关键符号: _fused_qk_rmsnorm_rope_gate_kernel,
fused_qk_gemma_rmsnorm_rope_gate, MRotaryEmbedding._build_axis_map,
MRotaryEmbedding._legacy_axis_map, Ernie4_5_VLRotaryEmbedding._build_axis_map,
Qwen3_5AttentionDecoderLayer.forward_prepare_cuda_fused

关键源码片段

python/sglang/srt/layers/rotary_embedding/mrope.py

axis map 统一构造是本修复的数据契约核心: 把 GLM-only 的 map 构造泛化到三种布局, 并提供 `_legacy_axis_map` 隔离新旧 kernel 的输入差异。

```
# 回答“每个 rotary lane 由哪个轴 (t/h/w) 拥有”的核心构造。
# 三种布局只是 lane 归属规则不同, 统一收敛到这里后,
# fused kernel 才能按轴取 mrope 位置。
def _build_axis_map(self) -> Optional[torch.Tensor]:
    """Which of the temporal, height and width axes owns each rotary lane."""
    if not self.mrope_section:
        return None
    section = self.mrope_section
    num_pairs = self.rotary_dim // 2
    assert (
        len(section) == 3 and sum(section) == num_pairs
    ), f'mrope_section {section} must be three axes summing to {num_pairs}'
    if self.mrope_interleaved_glm:
        # GLM 采用 round-robin, 轴配额耗尽后跳过; 其消费 kernel 在树外,
        # 因此顺序只能被测试 pinned, 无法直接对比。
        axes = []
        spent = [0, 0, 0]
        for lane in range(num_pairs):
            axis = lane % 3
            while spent[axis] >= section[axis]:
                axis = (axis + 1) % 3
            spent[axis] += 1
            axes.append(axis)
    elif self.mrope_interleaved:
        # Qwen 系 interleaved: t 先占满, h、w 按 3 步长交错填充。
        axes = [0] * num_pairs
        for axis in (1, 2):
            for lane in range(axis, min(3 * section[axis], num_pairs), 3):
                axes[lane] = axis
    else:
        # contiguous: 按 section 顺序平铺, 等价于一次 split + cat。
        axes = [axis for axis, size in enumerate(section) for _ in range(size)]
```

```
return torch.tensor(axes, dtype=torch.long, device=self.cos_sin_cache.device)
```

```
# 旧 kernel (triton_mrope_fused / sgl_kernel.multimodal_rotary_embedding)
# 只认识 GLM 的 round-robin, 非 GLM 必须继续收 None, 否则输入契约变化。
@property
def _legacy_axis_map(self) -> Optional[torch.Tensor]:
    """The map only where the older rope kernels read it; one is out of tree."""
    return self.axis_map if self.mrope_interleaved_glm else None
```

python/sglang/kernels/ops/attention/fused_qk_rmsnorm_rope_gate.py

bug 的实际发生地; kernel 按 axis map 加载 mrope 位置是修复主战场, 且入口新增 mrope 契约断言。

```
# 每个 rotary lane 先查 axis map, 确定它属于 t/h/w 哪一行,
# 再按“行 stride”定位该 token 在该轴上的位置。
# 行 stride 来自张量自身 (positions.stride(0)) 而不是常数 3,
# 因为 CUDA-graph decode 会从 [3, max_num_token] buffer 重放
# mrope_positions[:, :num_tokens] 的切片, 行距仍是 buffer 的行距。
# rot_offs 是当前 lane 的旋转下标, rot_mask 屏蔽 rotary_dim 之外的尾部。
if MROPE:
    axis = tl.load(mrope_axis_map_ptr + rot_offs, mask=rot_mask, other=0)
    pos = tl.load(
        positions_ptr + axis * stride_pos_axis + token,
        mask=rot_mask,
        other=0,
    )
else:
    pos = tl.load(positions_ptr + token)

cache_off = pos.to(tl.int64) * stride_cos_t
cos = tl.load(
    cos_sin_cache_ptr + cache_off + rot_offs, mask=rot_mask, other=0.0
).to(tl.float32)
```

评论区精华

PR 全程 `review_comments_count = 0`, 没有代码级争论。唯一审核意见来自 BBuf: APPROVED, 评价“Great fix, thanks.”。CI 层面 gongy 多次发起 `/rerun-test` (首次 dispatch 422 失败后重试成功), 覆盖 kernel、rotary、mrope 与 VLM 服务端共 8 个测试, 最终 1-gpu-h100 与 ubuntu-latest 全绿; 但 PR body 中 base 与 AMD ROCm 7.2 两格仍为红, 依赖 `bypass-fastfail` 标签放行。

- CI 复跑 mrope/rope/VLM 相关测试 (other): 1-gpu-h100 3 个测试与 ubuntu-latest 5 个测试全部通过; PR 合入时 base CI 与 AMD ROCm 7.2 仍为红, 靠 `bypass-fastfail` 放行。
- 唯一审核意见: 直接批准 (other): PR 合入 main。

风险与影响

- 风险：
 - 默认路径行为变化：Qwen3_5AttentionDecoderLayer.self_attention 在 `_is_cuda` and `attn_output_gate`（默认 `True`）下无条件走 `fused kernel`，因此本次修复会改变 CUDA 上所有带图请求的 RoPE 输出。这属于正确性修复，但与旧版本推理结果不一致，若有依赖旧输出的缓存或基线需要重建。
 - 布局隐含假设：`_build_axis_map` 隐含 `mrope_section` 按 `t`、`h`、`w` 排序；Ernie 通过覆盖规避，但未来新增模型若顺序不同而忘记覆盖，会静默使用错误的 `axis map`。
 - 旧 kernel 契约：`_legacy_axis_map` 只在 GLM 下返回 `map`。若未来有非 GLM 模型也需要把 `map` 传给 `triton_mrope_fused` 或 `sgl_kernel.multimodal_rotary_embedding`，当前设计会静默传 `None`，需要主动修改。
 - 连续性断言：入口要求 `positions.stride(1) == 1`，对 2-D `mrope` 输入是合理约束；若未来上层传入非连续切片会直接 `assert`，属于 `fail-fast`，风险可控。
 - 测试与 CI 覆盖缺口：新 kernel 测试注册在 1-gpu-large CUDA 上，HIP/XPU/CPU/NPU 没有针对本次修改的测试（这些分支走 `forward_native`，不受影响但未验证）；AMD ROCm 7.2 CI 保持红状态合入。
- 影响：
 - 正确性影响：CUDA 上 Qwen3_5ForConditionalGeneration 与 Qwen3_5MoeForConditionalGeneration 的所有带图请求，从“h/w 轴位置被丢弃”恢复为按真实 `mrope` 坐标旋转；文本-only 请求、HIP/XPU/CPU/NPU 分支、以及复用同一层的 `qwen3_5_mtp`、`minicpmv`、`interns2preview` 不受影响。
 - 推理结果变化：图像视觉定位与回答的边界框输出会变化（PR 端到端示例从固定 `x1 = 800` 变为随提问词移动，趋近 `unfused` 路径），相关多模态基准需要回归。
 - 性能影响：`mrope` 特化每个 `lane` 多一次 `int64` 位置加载，实测对 `prefill 3072 tokens` 与 `decode 32 tokens` 的 `launch` 时间影响约 $\pm 0.1\%$ ，可忽略。
 - 工程影响：`axis_map` 成为 `MRotaryEmbedding` 的稳定输出，后续其他 `fused kernel` 可以直接复用；`_legacy_axis_map` 为旧 kernel 划清了兼容边界，降低回归风险。
 - 风险标记：核心路径变更，默认路径行为变化，布局隐含假设，旧 kernel 契约，AMD 与 base CI 未全绿

关联脉络

- 暂无明显关联 PR