PR #34505 完整报告

sgl-project/sglang

[Diffusion][MiniMax H3] Extend exact QKNorm+RoPE rounding to SM103

合并时间：2026-08-12 16:25

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/34505

PR #34505 分析报告

执行摘要

PR #34505 是 MiniMax H3 Diffusion 内核的一次小规模位精确性修复：将 python/sglang/kernels/jit/csrc/diffusion/qknorm_rope.cuh 中精确 BF16 舍入逻辑的架构覆盖从仅 SM120 扩展到 SM103（B300）。动机是 nvcc 在 SM103 上同样会收缩打包的 rotary 表达式，导致融合结果与 split BF16 参考实现相差 1 ULP。改动仅 11 行、单文件，SM120+ 与非 Blackwell 路径完全不变，实测性能无回归，属低风险正确性补丁。

功能与动机

PR body 明确指出："exact QKNorm+ RoPE path added for SM120 also needs the explicit BF16 rounding helpers on B300 (SM103)"。没有这些 helper 时，nvcc 会 contracts/promotes 打包的 rotary 表达式，使融合结果与 split BF16 QKNorm + RoPE 参考实现相差 1 ULP。也就是说，该修复要解决的是跨 Blackwell 架构的数值一致性问题：SM120 已启用精确舍入，SM103 必须同样启用才能与参考实现 bit-exact 对齐。PR 还明确声明这是纯正确性修改，"Do not change the kernel launch or non-Blackwell code path"。

实现拆解

1. 定位改动点：确认 python/sglang/kernels/jit/csrc/diffusion/qknorm_rope.cuh 中三个 SGL_DEVICE 内联函数 rotary_mul_rn、rotary_add、rotary_sub 的精确舍入分支原先只在 __CUDA_ARCH__>= 1200 时编译。
2. 扩展条件编译：三处 #if 条件统一改为 __CUDA_ARCH__== 1030 || __CUDA_ARCH__>= 1200，使 SM103（B300）也进入位级 round-to-nearest 路径；SM120+ 行为保持不变，非 Blackwell 仍走默认融合乘加路径。
3. 同步注释：rotary_add 上方的说明从 "on SM120" 更新为 "on Blackwell SM103/SM120"，让后续维护者理解 nvcc 收缩问题适用的架构范围。
4. 验证配套：未新增测试文件，复用既有 test_qknorm_rope_preserves_split_bf16_rounding，在 B300 上以 torch.equal 同时校验 Q、K；并在生产形状 (7936, 56, 128)、rope dim 96、BF16 下给出性能对比（166.10us vs 166.55us，约 0.3% 噪声），确认无性能回归。
5. 影响面：仅影响 SM103 平台编译该 JIT 内核时的数值路径，kernel launch 配置、非 Blackwell 代码路径均未触碰。

python/sglang/kernels/jit/csrc/diffusion/qknorm_rope.cuh

唯一变更文件：三个 rotary 辅助函数（rotary_mul_rn、rotary_add、rotary_sub）的精确 BF16 舍入分支通过条件编译扩展覆盖 SM103（B300），修复融合结果与 split 参考实现相差 1 ULP 的问题。

```
// 文件：python/sglang/kernels/jit/csrc/diffusion/qknorm_rope.cuh // 本次变更：把精确 BF16 舍入辅助逻辑的启用范围从仅 SM120+ 扩展到 SM103（B300）。 // 背景：nvcc 在 Blackwell 上可能收缩打包的 rotary 表达式，导致融合结果与 // split BF16 QKNorm + RoPE 参考实现相差 1 ULP。 template<typenameT> SGL_DEVICETrotary_mul_rn(Tlhs,Trhs){ #if defined(__CUDA_ARCH__) && (__CUDA_ARCH__ == 1030 || __CUDA_ARCH__ >= 1200) // 精确路径：按 BF16 位模式分别对乘积做 round-to-nearest 舍入， // 确保与 split 参考实现的语义一致。 uint16_tlhs_bits; uint16_trhs_bits; ifconstexpr(std::is_same_v<T,bf16_t>){ // … 位级乘法与舍入原实现，本次仅放宽架构条件 } #endif // 其他架构仍走默认融合乘加路径，函数体未改动 } template<typenameT> SGL_DEVICETrotary_add(Tx,Tcos,Ty,Tsin){ #if defined(__CUDA_ARCH__) && (__CUDA_ARCH__ == 1030 || __CUDA_ARCH__ >= 1200) // nvcc 可能在 SM103/SM120 收缩打包表达式： // 先对两个乘积分别做精确 BF16 舍入，再逐位相加。 constTlhs=rotary_mul_rn(x,cos); constTrhs=rotary_mul_rn(y,sin); uint16_tlhs_bits; // … 位级加法与舍入原实现，本次仅更新注释与 guard #endif // 默认路径：直接 fused 乘加 } template<typenameT> SGL_DEVICETrotary_sub(Tx,Tcos,Ty,Tsin){ #if defined(__CUDA_ARCH__) && (__CUDA_ARCH__ == 1030 || __CUDA_ARCH__ >= 1200) constTlhs=rotary_mul_rn(x,cos); constTrhs=rotary_mul_rn(y,sin); uint16_tlhs_bits; // … 位级减法与舍入原实现，guard 与 rotary_add 同步扩展 #endif }
```

（说明：函数体内部位操作属于 SM120 时期既有实现，本次 diff 未涉及，故以注释占位。）

评论区精华

该 PR 没有 reviewer 讨论。唯一交互是作者 BBuf 的 CI 操作指令 /tag-and-rerun-ci，用于重新标记并触发测试。准确性结论（B300 上 torch.equal 通过、无保护时测试失败）均由作者在 PR body 中自测陈述，未出现设计权衡争论；PR Test (Extra) 当前显示失败状态，但评论中未记录失败原因。

风险与影响

- 平台覆盖窄：guard 精确写为 __CUDA_ARCH__== 1030 || __CUDA_ARCH__>= 1200，若未来出现同样受 nvcc 收缩影响的其它 Blackwell 变体（如 SM100/SM110），需要再次显式补充，存在可维护性成本。
- 数值正确性：该改动只是把已验证的 SM120 精确路径复制到 SM103，最坏情况是不生效并回到 1 ULP 偏差，不会引入新的错误；但依赖 nvcc 行为，不同 CUDA 版本需回归确认。
- 测试覆盖：验证依赖 B300 实物，常规 CI runner 未必覆盖；当前 Extra CI 失败的原因未见说明，建议确认是否与硬件 / 环境相关。
- 性能：实测 166.10us vs 166.55us，约 0.3% 差异为噪声，无实质回归。

关联脉络

- PR #34347 [Diffusion][MiniMax H3] Fix SM120 QKNorm+RoPE rounding 是本 PR 的直接前身，两者共享同一 qknorm_rope.cuh 与同一准确性测试；#34505 把 #34347 的精确舍入策略从 SM120 推广到 SM103。
- PR #34507 [Diffusion][Z-Image] Tune native QK RMSNorm launch for SM103 同属 SM103（B300）平台上的 Diffusion QK 相关内核调整，显示该平台内核的正确性与性能正被持续校准。
- 整体脉络：SGLang 的 Diffusion 内核维护正在系统性地覆盖 Blackwell 各架构（SM103/SM120）的数值精确性，此类“按架构枚举 guard”的模式预计会继续出现。