PR #34008 完整报告

sgl-project/sglang

[diffusion] GLM-Image bit-exact fused aten LayerNorm+modulate / qk-LN (H200 30-step denoise -8.1%)

合并时间：2026-08-08 13:26

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/34008

执行摘要

- 一句话：GLM-Image adaLN 链融合 Triton 内核，去噪提速 8.1%
- 推荐动作：值得精读。这是一个高质量的融合内核案例，展示了如何从 SASS 层面复现 aten 内核数值以保证 bit-exact 的完整方法论，包括 Welford 归约、inline PTX 倒数/rsqrt 近似、bf16 逐操作舍入以及运行时自校验回退机制。对于从事推理内核优化的工程师有直接借鉴价值，也体现了 diffusion 模型数值一致性的严格工程实践。

功能与动机

PR body 指出 GLM-Image 的 transformer block 仍以 eager 分段运行 adaLN 链：每 step 242 次裸 aten layer_norm 启动，qk-norm 在真实形状下两次启动耗时 301us，占 profiler 中 denoise GPU 时间的 10.1%；手写 modulate/gate 链再增加多轮 elementwise 内核。目标是融合为单 Triton 内核同时保持输出 bit-exact，使优化可默认启用而无需质量门控。

实现拆解

1. 新增 Triton 内核文件python/sglang/kernels/ops/diffusion/triton/layernorm_modulate.py：实现 fused_layernorm_modulate 和 fused_qk_head_layernorm 两个内核，分别完成 LN(x)*(1+scale)+shift 和逐 head 的 q/k LayerNorm。内核通过 inline PTX 精确复现 aten 的 vectorized_layer_norm_kernel 数值 DAG：Welford 逐线程串行累积、SHFL.DOWN 折叠树、倒数快速路径、rsqrt.approx 及 bf16 逐操作舍入。注册为 custom op 并提供 fake impl 以便 torch.compile 安全。
2. 模型侧包装器glm_image.py：新增 _glm_ln_modulate、_glm_qk_layernorm、_glm_residual_gate_add 三个包装函数，各自检查条件（CUDA、dtype、plain LayerNorm、形状对齐、是否已禁用等），调用融合内核，并在首次调用时用 torch.equal 与 eager 链比对；任何异常或失配都会永久禁用 fast path 并回退 eager。
3. 接入热点路径：GlmImageAdaLayerNormZero 的图像流和文本流分别调用 _glm_ln_modulate；GlmImageAdaLayerNormContinuous 同样接入；attention 前向中 q/k 同时存在时调用 _glm_qk_layernorm；两个 ff-gate 位置改用 _glm_residual_gate_add。原 eager 链保留为回退路径。
4. 测试配套：新增注册测试 test_glm_image_ln_modulate.py，参数化覆盖真实形状（1,4096,4096）、（1,4360,32,128）及文本流 / 部分 warp 形状，断言 torch.equal 位精确且 fast path 被实际启用（_GLM_FUSED_LN_MOD_VERIFIED 为 True），已注册到

base-b-kernel-unit CI 阶段。

关键文件：

- python/sglang/multimodal_gen/runtime/models/dits/glm_image.py（模块 模型层；类别 source；类型 data-contract；符号 _eager_ln_modulate, _glm_ln_modulate, _glm_qk_layernorm, _glm_residual_gate_add）：模型侧核心接线文件，新增三个 bit-exact 包装器并接入所有 adaLN/qk-norm/ff-gate 位置，包含运行时自校验与 fallback 逻辑。
- python/sglang/kernels/ops/diffusion/triton/layernorm_modulate.py（模块 内核层；类别 infra；类型 infrastructure；符号 _round_bf16_to_fp32, _rcp4, _div_rn, _rsqrt_approx）：核心 Triton 内核实现，复现 aten vectorized_layer_norm_kernel 的 SASS 级数值 DAG，实现 LN+modulate 和 qk-LN 融合。
- test/registered/kernels/ops/diffusion/test_glm_image_ln_modulate.py（模块 测试；类别 test；类型 test-coverage；符号 test_fused_ln_modulate_is_bit_exact, test_fused_qk_head_layernorm_is_bit_exact）：验证融合内核在真实形状下与 eager 链 torch.equal 位精确，并断言 fast path 实际被启用。

关键符号：_glm_ln_modulate, _glm_qk_layernorm, _glm_residual_gate_add, _eager_ln_modulate, fused_layernorm_modulate, fused_qk_head_layernorm, _welford_push, _welford_combine, _rcp4, _rsqrtf

关键源码片段

python/sglang/kernels/ops/diffusion/triton/layernorm_modulate.py

核心 Triton 内核实现，复现 aten vectorized_layer_norm_kernel 的 SASS 级数值 DAG，实现 LN+modulate 和 qk-LN 融合。

```
@triton.jit
def _rcp4(x):
    # nvcc 的倒数快速路径（对我们整数计数总是采用）。
    # 复现 MUFU.RCP + 两次 FFMA 修正的 4 指令序列。
    return tl.inline_asm_elementwise(
        asm="""{
        .reg .f32 r0, e, e2;
        rcp.approx.f32 r0, $1;
        fma.rn.f32 e, $1, r0, 0fBF800000;
        sub.ftz.f32 e2, 0f80000000, e;
        fma.rn.f32 $0, r0, e2, r0;
    }""",
        constraints="=f,f",
        args=[x],
        dtype=tl.float32,
        is_pure=True,
        pack=1,
    )

@triton.jit
```

```
def _welford_push(val, mean, m2, cnt, valid, MASKED: tl.constexpr):
    # ``valid`` 掩码屏蔽了未执行的 aten 线程（其状态必须保持不变）。
    # 序列与 nvcc 生成的 FFMA 顺序完全一致:
    # mean' = fma(delta, rcp(cnt+1), mean)
    # m2' = fma(delta, val - mean', m2)
    delta = val - mean
    new_cnt = cnt + 1.0
    recip = _rcp4(new_cnt)
    new_mean = tl.fma(delta, recip, mean)
    t = val - new_mean
    new_m2 = tl.fma(delta, t, m2)
    if MASKED:
        new_mean = tl.where(valid, new_mean, mean)
        new_m2 = tl.where(valid, new_m2, m2)
        new_cnt = tl.where(valid, new_cnt, cnt)
    return new_mean, new_m2, new_cnt
```

评论区精华

该 PR 没有 review 评论（comments_count=0, review_comments_count=0）。作者在 PR body 中详细阐述了数值复现方法和基准结果，核心决策包括：将位精确性作为默认路径的硬约束，采用运行时首次调用自校验 + 永久 fallback 的机制，而非依赖质量门控；同时强调 SASS 级逐指令验证是保证 bit-exact 的关键。

- 暂无高价值评论线程

风险与影响

- 风险:
  1. 平台相关位精确性：内核数值复现的是 torch 2.11 sm_90 的 vectorized_layer_norm_kernel，若 PyTorch 版本或 GPU 架构变化导致 aten 调度到不同实现，fast path 可能不再 bit-exact，但运行时自校验会自动 fallback，不会静默改变输出。
  2. 全局状态变量：_GLM_FUSED_LN_MOD_DISABLED 等模块级全局变量是进程级状态，虽只影响 GLM-Image 模型，但多实例或多模型并发推理时可能相互影响（概率极低）。
  3. 性能验证范围：基准只在 H200 上验证，其他 GPU（如 A100）上的收益可能不同，且 fallback 场景下无性能提升。
  4. head_dim 限制：qk 内核要求 head_dim <= 128 且 %4==0，虽然覆盖当前模型，但未来模型扩展到更大 head_dim 时需要扩展内核。 - 影响：影响范围集中于 GLM-Image 扩散模型的推理路径：默认配置下 DenoisingStage 提速 8.1%，端到端（扣除 AR 阶段）提升约 7.7%，且输出与 *eager* 完全一致（md5 相同），对用户完全无损。对团队而言，该方法论（SASS 级数值复现 + 自校验 fallback）可推广到其他 diffusion 模型或数值敏感内核，潜在推动更多 bit-exact 融合优化。 - 风险标记：平台相关位精确性，依赖 aten 内核调度，全局状态变量，仅 H200 验证

关联脉络

- PR #33558 [Laguna] fix YaRN mscale double-application in rope config: 同为本仓库数值一致性与 bit-exact 类修复，关注模型输出精度，与本 PR 的位精确目标一脉相承。
- PR #33417 Fix deterministic inference for Inkling: 同样是修复推理一致性问题，确保确定性输出，与本 PR 保证输出 bit-exact 的方法论相关。