PR #30883 完整报告

sgl-project/sglang

[XPU] Add qknorm_rope support for Flux

合并时间：2026-08-05 10:12

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/30883

执行摘要

- 一句话：XPU 为 FLUX.2 接入融合 QK-norm+RoPE，覆盖单流块
- 推荐动作：值得精读 layernorm.py 中「融合路径 gate 而非 assert」的 fallback 设计：CUDA/XPU 各自有 dtype、shape、contiguity 条件，不满足时优雅降级到原生路径，保证正确性优先。同时 review 中关于 Python 层逐点校验开销、@cache_once 与 C++ 校验下沉的权衡，以及『设备分支不应污染模型文件』的取舍，都是多平台推理框架开发的典型决策案例，建议一并阅读讨论记录。

功能与动机

FLUX.2 的 QK 规范化与 RoPE 原先在 XPU 上以分离算子执行，存在多次张量拷贝与 kernel 启动开销。PR body 明确目标：『Use fused_inplace_qknorm_rope for FLUX qknorm_rope to achieve better performance』，并指出单流块原先走 norm + 独立的 apply_flashinfer_rope_qk_inplace，约 40/56 个 QK-norm+RoPE 站点 / 步在 XPU 上落到了未融合路径。

实现拆解

整体实现分为 5 步：

1. 平台导入接入：在 python/sglang/multimodal_gen/runtime/layers/layernorm.py 的平台分支中，为 _is_xpu 增加 from sgl_kernel import fused_inplace_qknorm_rope，使 XPU 运行时能够解析融合 kernel 符号。
2. 新增 XPU fused 分支：在 apply_qk_norm_rope 的 CUDA fused 分支之后新增 _is_xpu 分支，条件限定在 allow_inplace、q_eps == k_eps、fp16/bf16、norm weight dtype 匹配、head_dim in (64, 128, 256) 与 rope_dim in (32, 64, 128, 256)；命中后原位更新 q/k 并直接返回。该分支不要求 q/k 完全连续，因为 XPU kernel 直接读取 token/head stride，只需最后维连续（chunk 视图天然满足），避免了模型侧拷贝。
3. 加固 CUDA fallback：apply_qk_norm 的 fused 分支新增 q.is_contiguous()/k.is_contiguous() 守卫（避免非连续输入在 q.view(batch_size, -1, head_dim) 处抛 view size is not compatible），fallback 从 .view(-1, head_dim) 改为 .reshape(-1, head_dim) 以兼容 strided 视图。这是单流块接入共享 helper 后触发的兼容性修复。

4. 调用侧统一：在 python/sglang/multimodal_gen/runtime/models/dits/flux_2.py 的 Flux2ParallelSelfAttention.forward（single-stream 块）中，将 norm_q/norm_k + apply_flashinfer_rope_qk_inplace 替换为 apply_qk_norm_with_optional_rope，并删除不再使用的 apply_flashinfer_rope_qk_inplace import，使约 40/56 个 QK-norm+RoPE 站点在 XPU 上进入 fused 路径。
5. 测试与 CI 配套：本 PR 未新增测试文件，精度覆盖依赖 sgl-kernel-xpu 侧测试；CI 中 amd-mi325 失败经 ck-intel 确认与本次改动无关（LongLive 模型 num_frames 配置校验，且 fused op 为 XPU gate，不会在 AMD/HIP runner 执行）。

关键文件：

- python/sglang/multimodal_gen/runtime/layers/layernorm.py（模块 归一化层；类别 source；类型 core-logic；符号 apply_qk_norm, apply_qk_norm_rope, apply_qk_norm_with_optional_rope）：核心变更文件：新增 XPU 的 fused_inplace_qknorm_rope 分支、加固 CUDA fused 路径的 contiguity 守卫，并将 fallback 从 view 改为 reshape 以兼容非连续输入。
- python/sglang/multimodal_gen/runtime/models/dits/flux_2.py（模块 扩散模型；类别 source；类型 data-contract；符号 Flux2ParallelSelfAttention.forward, apply_qk_norm_with_optional_rope）：调用侧接入：FLUX.2 单流注意力块从 norm + flashinfer rope 分离实现改为统一走 apply_qk_norm_with_optional_rope，使 XPU 融合路径覆盖约 40/56 个 QK-norm+RoPE 站点。

关键符号：apply_qk_norm, apply_qk_norm_rope, apply_qk_norm_with_optional_rope, Flux2ParallelSelfAttention.forward

关键源码片段

python/sglang/multimodal_gen/runtime/models/dits/flux_2.py

调用侧接入：FLUX.2 单流注意力块从 norm + flashinfer rope 分离实现改为统一走 apply_qk_norm_with_optional_rope，使 XPU 融合路径覆盖约 40/56 个 QK-norm+RoPE 站点。

```
# 文件：python/sglang/multimodal_gen/runtime/models/dits/flux_2.py
# Flux2ParallelSelfAttention.forward 中的 QK 规范化片段

        # qkv.chunk(...) 得到的是 last-dim 连续的视图；
        # unflatten 为 (heads, head_dim) 后 stride(-1) == 1 依然成立，
        # 因此 XPU 融合 kernel 可直接读取，无需模型侧拷贝。
        query = query.unflatten(-1, (self.local_heads, -1))
        key = key.unflatten(-1, (self.local_heads, -1))
        value = value.unflatten(-1, (self.local_heads, -1))

        cos_sin_cache = None
        if freqs_cis is not None:
            cos, sin = freqs_cis
            # 与双流块保持一致：cos/sin 拼成一张 2D cache，供 fused kernel 使用
            cos_sin_cache = torch.cat(
```

```
            [
                cos.to(dtype=torch.float32).contiguous(),
                sin.to(dtype=torch.float32).contiguous(),
            ],
            dim=-1,
        )

    # 单流块统一走共享 dispatcher：CUDA 命中 fused 路径，
    # XPU 命中 fused_inplace_qknorm_rope，不支持的形状自动回退，
    # 从而避免在模型文件里写设备相关的分支判断。
    query, key = apply_qk_norm_with_optional_rope(
        q=query,
        k=key,
        q_norm=self.norm_q,
        k_norm=self.norm_k,
        head_dim=self.head_dim,
        cos_sin_cache=cos_sin_cache,
        is_neox=False,
        allow_inplace=True,
    )
```

评论区精华

Review 核心交锋围绕三件事：

- CUDA 路径为何加 is_contiguous() 守卫：mingfeima 质疑『why we need to check this? this is cuda path.』；ck-intel 解释单流块接入共享 helper 后非连续（chunked）q/k 会进入 apply_qk_norm，没有守卫时 q.view(batch_size, -1, head_dim) 会抛『view size is not compatible』，守卫让连续输入走 fused、strided 输入走 fallback，不改变原 CUDA 行为。
- Python 层校验开销 vs C++ 校验：mingfeima 认为『this check is a little bit too long for python layer. each . in python will spend a tiny fraction of time』，建议把 head_dim/rope_dim/contiguity 校验放进 C++ kernel，不满足直接报错；CaoE 也补充『C++ has comprehensive checks. Retain only the basic checks here.』，最终 XPU 分支只保留 dtype 与 head_dim/rope_dim 集合等基本检查。
- 是否在模型 forward 里加 XPU 设备分支：CaoE 提议只在 Flux2ParallelSelfAttention.forward 加 XPU 检查调用融合路径；ck-intel 反对在模型文件放设备相关分支，并论证 CUDA 单流块行为几乎 perf-neutral（同一份 RMSNorm copy + flashinfer rope，只是调用路径不同），最终保持统一 dispatcher。
- 未来合并 CUDA/XPU 路径：CaoE 建议『Keep the status quo and rename it to fused_inplace_qknorm_rope for future merging with the CUDA path.』，代码中留有 TODO 注释，待 CUDA fused kernel 支持 last-dim 连续 q/k 后合并。
  - CUDA 路径新增 is_contiguous() 守卫的必要性 (correctness): 保留守卫：连续输入走 fused in-place kernel，strided 输入回退原生路径，CUDA 单流行为不变。
  - Python 层校验过长，校验下沉到 C++ kernel (design): 采纳：XPU 分支只保留 dtype 与 head_dim/rope_dim 集合等基本检查，C++ 完成全面校验。

- 用 @cache_once 减少重复检查开销 (performance): 最终按 mingfeima 意见简化为少量基本检查，未引入 cache_once。
- 是否在模型 forward 中加 XPU 设备分支 (design): 保持通用路径统一走 dispatcher，设备差异收敛在 layernorm.py。
- XPU 与 CUDA fused 路径未来合并 (design): 合并暂缓，以 TODO 注释记录：待 CUDA fused_inplace_qknorm_rope 支持 last-dim 连续 q/k 后合并。

风险与影响

- 风险:
  1. CUDA 共享路径回归风险：flux_2.py 单流块从 norm + flashinfer rope 改为走 apply_qk_norm_with_optional_rope，虽经论证语义等价（fallback 中 reshape 的拷贝与原 RMSNorm 内部拷贝一致），但属于共享推理路径变更，需要 CUDA 侧回归验证；CI 已覆盖相关用例。
  2. XPU 分支缺少 contiguity 守卫：头版本 XPU 分支条件中不再有 stride(-1)==1 检查，若未来调用方传入 head-dim 非连续张量，将依赖 C++ kernel 的校验直接报错（mingfeima 明确要求『just report error in C++』），行为从 fallback 变为 hard error，存在隐式契约风险。
  3. 缺少直接单测覆盖：本 PR 没有新增仓库内单元测试，精度依赖外部 sgl-kernel-xpu 测试，回归防线较弱。
  4. fallback reshape 拷贝：非连续输入下 reshape(-1, head_dim) 触发一次拷贝，与旧行为一致无额外开销；但若未来 CUDA fused kernel 支持 last-dim contiguous，性能还可进一步提升（代码已留 TODO）。
- 影响：影响面集中在 diffusion 多模态生成管线:
- 用户侧：XPU 上 FLUX.2 推理性能提升，约 40/56 个 QK-norm+RoPE 站点 / 步从分离算子变为融合 kernel；CUDA 用户行为不变，但获得了非连续输入兼容性修复。
- 系统侧：apply_qk_norm/apply_qk_norm_rope 是共享基础设施，FLUX.2 双流块与单流块现在统一走同一 dispatcher，设备差异收敛在 layernorm.py，后续新增平台（如 CUDA 合并分支）只需改一处。
- 团队侧：补强了 XPU 平台在 diffusion 场景的竞争力，与既有的 XPU 融合 kernel 战略（如 #28040）形成连续演进。
- 风险标记：共享推理路径变更，缺少直接单测覆盖，平台分支依赖 C++ 校验，CUDA 与 XPU 分支待合并

关联脉络

- PR #28040 [Intel GPU] DeepSeek V4 8/N: use sgl-kernel implementation of fused_k_norm_rope_flashmla on XPU: 同属 XPU 平台归一化 + RoPE 融合 kernel 落地路线，使用同一批 sgl-kernel XPU 算子。
- PR #33536 [diffusion] Fuse DiT FFN tanh-GELU into up-proj GEMM (cublasLt epilogue) behind quality=high: 同一 diffusion 推理加速主题，同样以 fused kernel 减少 kernel 启动与中间张量。

- PR #33451 [diffusion] FLUX.2 VAE decoder fast path behind quality=high: 同一 FLUX.2 性能优化系列，均围绕 sglang/multimodal_gen 运行时做算子融合与快速路径。
- PR #33575 [rotary] Rebuild the shared RoPE cache entry when its buffers are dead: 涉及 rotary_embedding 缓存正确性，与本 PR 共用 apply_flashinfer_rope_qk_inplace 基础设施。