PR #34126 完整报告

sgl-project/sglang

[diffusion] FLUX.1: route the adaLN LN+modulate sites through the bit-exact fused LayerNorm+modulate kernel (H200 1024^2 lossless denoise -1.2%, e2e wall -2.9%)

合并时间：2026-08-09 09:52

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/34126

执行摘要

- 一句话：FLUX.1 融合 LN+modulate 内核上线 lossless，提速 1.2%
- 推荐动作：值得精读。它本身改动很小（只 rewire 一个文件），但集中体现了 sglang diffusion 侧的性能工程方法论：quality tier 与 bit-exact 语义如何协同、运行时自验证如何让内核接入保持安全、以及如何用 md5/PSNR 协议量化每一步优化。建议结合 #34004、#34008、#33819 串读，可以完整看到 FLUX.1 优化从 plumbing 到 kernel 再到 wiring 的演进。

功能与动机

34004 刻意未将 LayerNorm 归约纳入 lossless 层级：当时逐位复现 torch 2.11 的 vectorized_layer_norm_kernel（per-element Welford、count-weighted cuWelfordCombine、多指令 rsqrtf、FMA 收缩选择）被认为风险高，所以 lossless 路径保留 aten LN + 融合 modulate（两个内核、三次 HBM 往返）。#34008 已将该复现做成可复用 Triton 内核并在 GLM-Image 上验证，本 PR 将其接入 FLUX.1 全部 adaLN 站点，让 lossless 层级在保持 bit-exact 的同时提速。

实现拆解

1. 引入内核与守卫（python/sglang/multimodal_gen/runtime/models/dits/flux.py）：新增对 sglang.kernels.ops.diffusion.triton.layernorm_modulate 的 can_use_fused_layernorm_modulate / fused_layernorm_modulate / is_plain_layer_norm 导入，并增加模块级变量 _FLUX_FUSED_LN_MOD_DISABLED 与 _FLUX_FUSED_LN_MOD_VERIFIED，分别承担全局禁用标志与已验证签名集合。
2. 新增 _flux_fused_ln_modulate 路由函数：先做静态守卫（未禁用、无 affine 的 LayerNorm、内核契约），再按 (shape, stride, eps) 计算签名；未验证签名在 torch.compile tracing 与 CUDA graph capture 中直接跳过（避免在 tracing 里执行 eager 链与 host sync）；正常执行时调用内核并与 eager 链

modulate_scale_shift(norm(x), scale, shift) 做 torch.equal 核对，通过则加入签名集合，任一签名失配即永久禁用并返回 *eager* 结果。

3. 重排 _flux_norm_modulate 优先级：依次为 (1) bit-exact 融合内核；(2) quality="high" 的 LN-affine fold（仅在内核不适用或验证失败时可达）；(3) aten LN + 融合 modulate（#34004 lossless 路径）。五个站点类（dual-stream norm1 / norm1_context / norm2 / norm2_context + single-stream）通过既有管线自动继承，Nunchaku 分支不动。
4. 测试配套：新增 test/registered/kernels/ops/diffusion/test_flux_ln_modulate.py（75 行，注册到 base-b-kernel-unit 1-gpu-large）。参数化四种真实站点形状的 bit-exact 断言（含 CFG batch 与非整 token 数）、bit-exact 优先于 high fold 的优先级检查、以及 hidden % 4 != 0 的 guard 拒绝；同时确认现有 #34004/#34008 相关套件（21 个测试）不改动且通过。
5. 验证与基准：H200 按 #33451/#33536/#33819 协议（seed 42、50 步、10 次生成去首帧）对比，lossless 层两次独立运行各 18 个样本，DenoisingStage 平均 -1.2%、服务端 -1.5%、e2e wall -2.9%；40 张输出图共享单一 md5 且与 #34004 基线相同。

关键文件：

- python/sglang/multimodal_gen/runtime/models/dits/flux.py（模块 扩散模型；类别 source；类型 core-logic；符号 _flux_fused_ln_modulate, _flux_norm_modulate）：模型侧唯一改动文件：新增 _flux_fused_ln_modulate 路由与 per-signature 首见验证，并重排 _flux_norm_modulate 优先级，让 bit-exact 融合内核在 lossless 默认路径生效。
- test/registered/kernels/ops/diffusion/test_flux_ln_modulate.py（模块 内核测试；类别 test；类型 test-coverage；符号 _eager, _make_site_inputs, test_flux_fused_ln_modulate_is_bit_exact, test_flux_norm_modulate_bitexact_supersedes_high_fold）：新增单测覆盖所有 FLUX.1 站点签名（dual/text/concat/CFG batch）的 bit-exact 验证、bit-exact 优先于 high fold 的优先级，以及内核契约的 guard 拒绝，是保证默认路径不变的关键证据。

关键符号：_flux_fused_ln_modulate, _flux_norm_modulate

关键源码片段

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

模型侧唯一改动文件：新增 _flux_fused_ln_modulate 路由与 per-signature 首见验证，并重排 _flux_norm_modulate 优先级，让 bit-exact 融合内核在 lossless 默认路径生效。

```
# flux.py: FLUX.1 adaLN 站点的 bit-exact 融合 LN+modulate 路由
# 模块级状态：一个全局禁用标志 + 一个已验证签名集合
_FLUX_FUSED_LN_MOD_DISABLED = False
# 已验证的 (shape, stride, eps) 签名：bit-exact 是 live aten dispatch 的属性，
# 所以每个新签名都要在运行时与 eager 链核对一次
_FLUX_FUSED_LN_MOD_VERIFIED: set = set()


def _flux_fused_ln_modulate(
    norm: nn.Module,
```

```
    x: torch.Tensor,
    scale: torch.Tensor,
    shift: torch.Tensor,
) -> Optional[torch.Tensor]:
    """单内核 ``LN(x) * (1 + scale) + shift``，与 eager 链 bit-exact，否则返回 None。

    该 Triton 内核复现了当前 dispatch 为 bf16 行选中的 aten LayerNorm 内核
    （PR #34008），但 bit-exact 是运行时属性：每个不同的 (shape, stride, eps)
    组合首次出现时都会与 eager 链做 torch.equal 核对，任何失配就永久禁用
    fast path 并回退 eager。
    """
    global _FLUX_FUSED_LN_MOD_DISABLED

    # 静态守卫：全局未禁用、LayerNorm 无 affine 参数、dtype/layout/ 形状满足内核契约
    if (
        _FLUX_FUSED_LN_MOD_DISABLED
        or not is_plain_layer_norm(norm, x.shape[-1])
        or not can_use_fused_layernorm_modulate(x, scale, shift)
    ):
        return None
    sig = (
        x.shape,
        x.stride(),
        scale.shape,
        scale.stride(),
        shift.shape,
        shift.stride(),
        norm.eps,
    )
    verified = sig in _FLUX_FUSED_LN_MOD_VERIFIED
    if not verified and (
        torch.compiler.is_compiling() or torch.cuda.is_current_stream_capturing()
    ):
        # 首见核对需要 eager 链和 host sync，不能在编译追踪或 CUDA graph
        # 捕获期间执行，此时直接放弃 fast path（warmup 阶段会先完成验证）
        return None
    try:
        out = fused_layernorm_modulate(x, scale, shift, norm.eps)
    except Exception as exc:
        if torch.compiler.is_compiling():
            raise
        logger.warning_once(f"Disabling FLUX fused LN+modulate fast path: {exc}")
        _FLUX_FUSED_LN_MOD_DISABLED = True
        return None
    if verified:
        return out
    ref = modulate_scale_shift(norm(x), scale, shift)
    if torch.equal(out, ref):
        _FLUX_FUSED_LN_MOD_VERIFIED.add(sig)
```

```
        return out
    logger.warning_once(
        "FLUX fused LN+modulate fast path is not bit-exact against this "
        "platform's LayerNorm dispatch; falling back to eager"
    )
    _FLUX_FUSED_LN_MOD_DISABLED = True
    return ref


def _flux_norm_modulate(
    site: nn.Module,
    norm: nn.Module,
    x: torch.Tensor,
    scale: torch.Tensor,
    shift: torch.Tensor,
) -> torch.Tensor:
    """``norm(x) * (1 + scale) + shift`` 的三级优先路由。

    优先级：(1) bit-exact 单内核 LN+modulate，lossless 且无质量闸门；
    (2) quality="high" 的 LN-affine fold，仅在内核不适用或验证失败时可达；
    (3) aten LN + bit-exact 融合 modulate（#34004 的 lossless 路径）。
    """
    out = _flux_fused_ln_modulate(norm, x, scale, shift)
    if out is not None:
        return out
    if fused_ln_modulate_active(site) and can_fuse_ln_modulate(x, scale, shift):
        return fused_ln_modulate(x, scale, shift, norm.eps)
    return modulate_scale_shift(norm(x), scale, shift)
```

test/registered/kernels/ops/diffusion/test_flux_ln_modulate.py

新增单测覆盖所有 FLUX.1 站点签名（dual/text/concat/CFG batch）的 bit-exact 验证、bit-exact 优先于 high fold 的优先级，以及内核契约的 guard 拒绝，是保证默认路径不变的关键证据。

```
# test_flux_ln_modulate.py：确保 FLUX.1 融合 LN+modulate 快路径与 eager 链 bit-exact
def _eager(norm, x, scale, shift):
    # eager 参考：norm 后做 (1 + scale) 缩放与 shift 平移
    return norm(x) * (1 + scale[:, None]) + shift[:, None]


def _make_site_inputs(shape, chunks, seed):
    # 按真实 FLUX.1 站点构造输入：x 是 bf16 激活，scale/shift 是 adaLN
    # 投影经 chunk 切出的 stride 视图，模拟 chunk(6)/chunk(3) 的布局
    torch.manual_seed(seed)
    batch, seq, hidden = shape
    norm = torch.nn.LayerNorm(hidden, eps=1e-6, elementwise_affine=False).cuda()
    x = (torch.randn(batch, seq, hidden, device="cuda") * 8).bfloat16()
    emb = torch.randn(batch, chunks * hidden, device="cuda").bfloat16()
    parts = emb.chunk(chunks, dim=1)
```

```
    return norm, x, parts[0], parts[1]


@pytest.mark.parametrize(
    "shape,chunks",
    [
        ((1, 4096, 3072), 6), # dual-stream 图像 tokens（1024^2），chunk(6)
        ((1, 512, 3072), 6), # dual-stream 文本 tokens
        ((1, 4608, 3072), 3), # single-stream 拼接，chunk(3)
        ((2, 300, 3072), 6), # CFG batch、非整 token 数
    ],
)
def test_flux_fused_ln_modulate_is_bit_exact(shape, chunks):
    # 每个 FLUX.1 站点会发出的 (shape, stride, eps) 签名都必须首次即验证通过
    norm, x, shift, scale = _make_site_inputs(shape, chunks, seed=0)
    out = _flux_fused_ln_modulate(norm, x, scale, shift)
    assert out is not None # 快路径必须被触发
    assert torch.equal(out, _eager(norm, x, scale, shift))
    assert not flux._FLUX_FUSED_LN_MOD_DISABLED
    assert flux._FLUX_FUSED_LN_MOD_VERIFIED
```

评论区精华

本 PR 无 reviewer 评论（review_comments_count=0），讨论主要来自作者在 body 与 issue 评论中的确认。核心论点是：与 #34004 的 quality="high" LN-affine fold 的关系——作者实测两种终态（a. 删除 fold；b. 保留为 fallback）在 H200 上执行完全一致，最终选择保留 fallback 以保护其他 torch 构建 /GPU 上的 quality="high"，代价是约 40 行已合并代码；若维护者倾向删除，可在 #34004 中 drop commit 3。另一个重点是按 (shape, stride, eps) 签名做首见验证的设计，相比 #34008 的单个全局标志更精确，但也说明 bit-exact 是 live aten dispatch 的属性而非内核本身的属性。

- quality="high" 的 LN-affine fold 应该删除还是保留为 fallback (design): 采用方案 b：保留 fold 作为 fallback，保护其他 torch 构建 /GPU 上的 quality="high"；若维护者倾向删除，可在 #34004 中 drop commit 3。
- lossless 层级在 rebase 后仍保持端到端 bit-exact (correctness): 默认路径 byte-exact 得到确认，可安全合并。
- CI 中非必需 red check 与本 PR 无关 (other): 必需检查 14/14 + lint/gate/check-changes 全绿，不阻塞合并。

风险与影响

- 风险：核心生成路径变更：FLUX.1 每步 114 个 adaLN 站点全部改走新路由，任何误判都会影响每张图的输出；但运行时 torch.equal 验证 + 永久回退将风险限制在首见签名的性能上。平台与版本耦合：bit-exact 只对当前 aten dispatch 成立，其他 torch 构建或 GPU 可能触发 SASS 复现失效，此时 fast path 会被禁用，功能不受影响，但 quality="high" 的 fold fallback 分支（约 40 行）在 H200 上成为死代码，维护成本上升。首见验证开销：每个新签名首次出现会多一次 eager 链计算与 host sync，且验证被刻意排除在 CUDA

graph capture 之外；若 warmup 未覆盖全部签名，capture 后的首见签名会静默回退 eager，可能带来一次意外的慢调用。测试盲区：新增单测覆盖 hidden % 4 != 0 的拒绝，但未覆盖 hidden > 8192 的上界，Nunchaku/ 量化分支也未测试（虽然它们不经过该路径）。

- 影响：用户：FLUX.1 默认 lossless 路径输出不变（md5 逐位一致）但更快（H200 每图服务端 -1.5%、DenoisingStage -1.2%）；quality="high" 用户服务端 -0.8%，且 high 与 lossless 的图像差异进一步收窄（PSNR 35.5 -> 34.7 dB），一致性更好。系统：每步减少一次内核发射和一次 HBM 往返（[1,L,3072] 激活），多请求并发下带宽压力降低。团队：确立了 "bit-exact 内核 + 按签名首见验证 + 永久回退 " 的接入模式，为后续把类似内核推广到其他 DiT/ 模型提供了可复用范式；同时为 FLUX 进一步把 linear+GELU epilogue 也纳入 lossless 层级铺路（目前 GELU epilogue 仍只属于 quality="high"）。
- 风险标记：核心生成路径变更，内核平台适用性依赖，首见验证引入额外开销，quality=high fold 遗留死代码

关联脉络

- PR #34015 [diffusion] Sana: bit-exact fused aten LayerNorm+modulate under BCG (H200 denoise -4.8%): 本 PR 使用的 fused_layernorm_modulate 内核由 #34008 引入并在 GLM-Image 验证，#34015 是该内核在 Sana 上的另一应用；本 PR 的单测契约也跟随 #34015 对内核的泛化（hidden % 4 == 0、上限 8192）。
- PR #34085 [diffusion] Clean up kernels and shared fast paths: 本 PR rebase 后适配 #34085：eager 参考从被内联掉的 _flux_modulate 改为 modulate_scale_shift(norm(x), scale, shift) 包装，并随之更新单测断言。
- PR #33819 [diffusion] FLUX.1 bit-exact residual-gate fast path + tanh-GELU epilogue behind quality=high (H200 e2e -1.1% lossless / -4.3% high): 同一文件的上一轮 FLUX.1 优化，建立了 mount/unmount 协议与 benchmark 协议（seed 42、50 步、md5 校验），本 PR 沿用并深化其 bit-exact 验证方法论。
- PR #33536 [diffusion] Fuse DiT FFN tanh-GELU into up-proj GEMM (cublasLt epilogue) behind quality=high (Qwen-Image 1024^2 denoise 12.36 -> 12.05 s on H200): quality tier 体系与 GELU epilogue 协议的早期落地；本 PR 使得 FLUX.1 quality=high 与 lossless 的差异进一步收窄（仅剩 linear+GELU epilogue）。