PR #33823 完整报告

sgl-project/sglang

[diffusion] FLUX.2 bit-exact residual-gate fast path (H200 klein-4B 50-step denoise -1.2%)

合并时间：2026-08-06 22:54

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/33823

执行摘要

- 一句话：FLUX.2 接入 bit-exact 残差门控融合，denoise 提速 1.2%
- 推荐动作：值得精读。重点看三点：一是 _flux2_residual_gate_add 的降级语义设计（一次性禁用 + 编译期重抛），这是「复用内核但绝不暗中改变语义」的范本；二是 bit-exact 验证协议的层次结构（kernel 级 torch.equal → 整图 md5 → 多轮交替 A/B 归因），可作为性能优化 PR 的验收模板；三是与 quality=high 门控的边界划分——bit-exactness 是优化能否安全落在默认路径上的关键判据。如果只关心结果，可跳过测试文件细节。

功能与动机

FLUX.2 的 transformer 通过 *eager* 两内核 gate * update + residual + … 对更新残差流，与 FLUX.1（#33819）和 ERNIE-Image（#33734）是同一模式：FLUX.2-dev 有 8 dual-stream + 48 single-stream blocks（hidden 6144），每步 80 个 gate 位置，50 步共 4,000 对 *eager* 乘加；klein-4B 每步 40 处、50 步 2,000 对。而 residual_gate_add_cuda 内核自 #29361 就在树内且已在 LTX-2 默认路径上使用，half dtype 下可逐位复现 *eager* pair 的舍入，因此可以在 lossless 默认路径上直接启用，不需要 quality 门控，也不会改变输出分布。

实现拆解

1. 新增 helper 与内核算力接入：在 python/sglang/multimodal_gen/runtime/models/dits/flux_2.py 顶部导入 *can_use_residual_gate_add_cuda* 与 residual_gate_add_cuda，新增模块级函数 _flux2_residual_gate_add 和全局开关 _FLUX2_RESIDUAL_GATE_CUDA_DISABLED。helper 的语义是「条件满足走内核，否则退回 *eager* 参照实现」，与 #33734/#33819 的接入形态完全一致。
2. 接线 5 个残差门控位置：Flux2SingleTransformerBlock.forward 的 joint 流（img+txt 拼接）1 处，Flux2TransformerBlock.forward 的 *image* 流 attn + MLP 与 *text* 流 attn + MLP 共 4 处，全部从 *residual + gate * update eager* 表达式替换为 _flux2_residual_gate_add 调用。FLUX.2 的调制参数天然是稠密 [1, 1, D] 行，无需像 ERNIE-Image 那样额外做 *contiguous* 预处理。
3. 约束与降级语义：内核只在 fp16/bf16 下 bit-exact，fp32 保持 *eager*（避免 fma 收缩破坏舍入）；批量 [B>1, 1, D] 由 *can_use_residual_gate_add_cuda* 拒绝后自动降级；运行时异常通过全局开关一次性禁用，且 torch.compiler.is_compiling() 为真时直接重抛，防止编译图静默 bake 住 fallback——这是 torch.compile 路径不受影响的保证。

4. 测试配套：test/registered/kernels/ops/diffusion/test_residual_gate_add.py 在 CASES 中补充 FLUX.2-klein（D=3072）与 FLUX.2-dev（D=6144）的真实 1024^2 shape（如 (1, 4608, 6144)），沿用半精度 atol=0/rtol=0 逐位断言；PR body 报告 16 passed。验证协议还包括整图 md5 一致（50 步与 4 步 preset 均一致，共 10 次迭代）以及 kernel 级 microbenchmark 归因（klein 40 sites 合计 1.04 ms/step）。

关键文件：

- python/sglang/multimodal_gen/runtime/models/dits/flux_2.py（模块 DiT 模型；类别 source；类型 core-logic；符号 _flux2_residual_gate_add, Flux2SingleTransformerBlock.forward, Flux2TransformerBlock.forward）：核心变更文件：新增 _flux2_residual_gate_add helper 并接入 5 个残差门控位置（Flux2SingleTransformerBlock x 1 + Flux2TransformerBlock x 4），half dtype 下与 eager 逐位一致，直接落在 lossless 默认路径，内核零改动。
- test/registered/kernels/ops/diffusion/test_residual_gate_add.py（模块 内核测试；类别 test；类型 test-coverage）：扩展内核位级一致性测试：补充 FLUX.2-klein（D=3072）与 FLUX.2-dev（D=6144）的真实 1024^2 shape，确保半精度下 atol=0/rtol=0 的逐位断言覆盖新接入的 shape 域。

关键符号：_flux2_residual_gate_add

关键源码片段

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

核心变更文件：新增 _flux2_residual_gate_add helper 并接入 5 个残差门控位置（Flux2SingleTransformerBlock x 1 + Flux2TransformerBlock x 4），half dtype 下与 eager 逐位一致，直接落在 lossless 默认路径，内核零改动。

```
# flux_2.py —— FLUX.2 DiT 残差门控融合入口
#
# 背景：FLUX.2 每个 transformer block 都用 eager 两内核
# `residual + gate * update` 更新残差流。dev 规模下每步 80 个
# gate 位置（8 dual x 4 + 48 single x 1），50 步就是 4,000 对
# 分离的乘加内核发射。
#
# 复用自 #29361 的 residual_gate_add_cuda 内核（LTX-2 默认路径
# 已在用）：half dtype 下它与 eager 两步舍入逐位一致，因此可以
# 直接进入 lossless 默认路径，不需要 quality="high" 门控。

from sglang.kernels.ops.diffusion.residual_gate_add import (
    can_use_residual_gate_add_cuda,
    residual_gate_add_cuda,
)

# 模块级禁用开关：内核首次运行异常后置位，后续调用全部走 eager，
# 避免每次请求都反复触发异常处理（一次性 fallback 语义）。
_FLUX2_RESIDUAL_GATE_CUDA_DISABLED = False

def _flux2_residual_gate_add(
```

```
    residual: torch.Tensor,
    update: torch.Tensor,
    gate: torch.Tensor,
) -> torch.Tensor:
    """单内核完成 `residual + gate * update`，与 eager 参照实现逐位一致。

    限定 half dtype：fp32 下内核会收缩成一次 fma（两步舍入变一步），
    破坏 bit-exact 契约，因此 fp32 必须留在 eager 路径。
    内核的按行广播 gate 只支持 `[1, ..., 1, D]`；批量 `[B>1, 1, D]`
    会让 `can_use_residual_gate_add_cuda` 拒绝并自动退回 eager。
    """
    global _FLUX2_RESIDUAL_GATE_CUDA_DISABLED

    if (
        not _FLUX2_RESIDUAL_GATE_CUDA_DISABLED
        and residual.dtype in (torch.float16, torch.bfloat16)
        and can_use_residual_gate_add_cuda(residual, update, gate)
    ):
        try:
            return residual_gate_add_cuda(residual, update, gate)
        except Exception as exc:
            # 编译期不吞异常：若 torch.compile 图里 bake 住 fallback，
            # 会导致每次图执行都携带异常路径语义且不可见。
            if torch.compiler.is_compiling():
                raise
            logger.warning_once(f"Disabling FLUX.2 residual-gate CUDA fast path: {exc}")
            _FLUX2_RESIDUAL_GATE_CUDA_DISABLED = True

    # eager 参照实现：与 diffusers 原始语义完全一致
    return residual + gate * update

# Flux2SingleTransformerBlock.forward：联合 img+txt 流只有 1 个 gate 位置
hidden_states = _flux2_residual_gate_add(hidden_states, attn_output, mod_gate)

# Flux2TransformerBlock.forward：img/txt 双流各 2 个 gate 位置（attn/mlp），
# 文本流与图像流共用同一 helper，保证两条流的舍入行为一致
hidden_states = _flux2_residual_gate_add(hidden_states, attn_output, gate_msa)
hidden_states = _flux2_residual_gate_add(hidden_states, ff_output, gate_mlp)
encoder_hidden_states = _flux2_residual_gate_add(
    encoder_hidden_states, context_attn_output, c_gate_msa
)
encoder_hidden_states = _flux2_residual_gate_add(
    encoder_hidden_states, context_ff_output, c_gate_mlp
)
```

评论区精华

本 PR 没有 reviewer 评论（review_comments 为空），唯一 comment 是作者 BBuf 发布的 CI 运行链接，因此没有多方交锋记录。值得记录的权衡来自 PR body 的两处「主动放弃」：

- norm+scale/shift rider 被丢弃：唯一 bit-exact 的共享实现 norm_scale_shift_native 硬性限定 Blackwell（capability >= 10），而本验证环境是 H200/SM90，与 #33819 放弃的理由相同。
- FLUX.1 的 GELU epilogue companion 不适用：FLUX.2 的 FFN 是 SwiGLU（Flux2SwiGLU），up-proj 已并入 QKV 投影，GELU 融合没有可挂载的位置。 这两处放弃与 #33734 对已关闭 PR #30170 三个内核逐个复验的处置一脉相承：只把「证明过 bit-exact」的优化放上默认路径，其余一律不碰。
- 暂无高价值评论线程

风险与影响

- 风险：
  - 默认路径语义变更：这是 lossless 默认路径上行为可观测的改动（虽已做 bit-exact 证明）。证明覆盖范围是 H200 + bf16/fp16 的 kernel 级 torch.equal（atol=0）与 bf16 整图 md5；fp16 只做了 kernel 级验证，未做整图 md5，其他 NVIDIA 平台依赖内核自身实现一致性。
  - 降级路径无单测：_FLUX2_RESIDUAL_GATE_CUDA_DISABLED 异常禁用、torch.compiler.is_compiling() 重抛、can_use_residual_gate_add_cuda 拒绝等 fallback 分支没有对应的单元测试，只能靠集成环境兜底。
  - 平台依赖：residual_gate_add_cuda 是 CUDA 内核，非 CUDA 平台（AMD/XPU）永远走 eager fallback，虽无行为风险但也没有收益；未来若在更多平台接入该内核，需重新验证 bit-exact 契约。
  - 小 shape 负优化：microbenchmark 中 (1, 512, 3072) 融合后 13.0 us 略慢于 eager 的 12.1 us（0.93x），但仍走融合路径；klein 整图实测为 -1.2% 正收益，但极端小 batch 或短 prompt 场景值得留意。
  - 批量支持缺口：[B>1, 1, D] gate 会静默降级到 eager，当前 FLUX.2 推理以 B=1 为主，若未来开启批处理需重新审视 gate 布局。
- 影响：
  - 用户侧：FLUX.2 全系列（klein 与 dev）请求在默认 lossless 路径上自动提速，输出与 main 位级一致（md5 验证），无 opt-in、无 API 变化。
  - 系统侧：每步减少 40–80 次内核发射与中间 HBM 往返；H200 klein-4B 50 步 denoise 3.358 → 3.318 s（-1.2%），dev 基于 D=6144 实测行推算约 -215 ms/ 图。
  - 团队侧：与 #33734、#33819 一起确立了「bit-exact 融合 → lossless 默认路径；非 bit-exact → quality=high 门控」的扩散推理优化范式，后续新模型（如 Wan、其他 DiT）适配可直接复用该 helper 模板。
  - 风险标记：默认路径变更，fallback 路径缺少测试覆盖，内核平台依赖，小 shape 负优化

关联脉络

- 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): 本 PR 的直接模板：同一

helper 形态、同一验证协议（torch.equal + 整图 md5）；FLUX.1 额外挂了quality=high 的 GELU epilogue，而 FLUX.2 因 SwiGLU 不适用，只做 gate 接线。

- PR #33734 [diffusion] ERNIE-Image bit-exact residual-gate fast path (H200 1024^2 e2e 16.17 -> 15.75 s): 确立 kernel 级 torch.equal + 整图 md5 的 bit-exact 判据，并对已关闭 PR #30170 做逐个内核复验；本 PR 沿用其 helper 与验证协议。
- PR #33451 [diffusion] FLUX.2 VAE decoder fast path behind quality=high (H200: 1024^2 97.6->29.2 ms, 2048^2 437.2->168.5 ms): FLUX.2 解码侧优化（quality=high 门控），本 PR 是 DiT 侧优化（lossless 默认路径），两者互补构成 FLUX.2 全链路加速，PR body 明确声明相互独立。
- 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=high 门控与 mount/unmount 协议来源；本 PR 因内核 bit-exact 选择 lossless 默认路径，与 #33536 的非 bit-exact 门控形成「bit-exact 与否决定路径归属」的方法论对照。