PR #33819 完整报告

sgl-project/sglang

[diffusion] FLUX.1 bit-exact residual-gate fast path + tanh-GELU epilogue behind quality=high (H200 e2e -1.1% lossless / -4.3% high)

合并时间：2026-08-06 19:56

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/33819

执行摘要

- 一句话：FLUX.1 接入 bit-exact 残差门控与 quality=high GELU 融合，lossless 提速 1.1%、high 提速 4.3%
- 推荐动作：值得精读。该 PR 展示了如何在保持默认 bit-exact 的同时，借助 quality 分层接入非 bit-exact 优化；_flux_residual_gate_add 的 fallback、编译逃逸与位精确验证设计是同类融合的优秀范本。

功能与动机

PR 正文指出 FLUX.1 每张 1024x1024 图共执行 5,700 个 eager 双内核残差门控对和 3,800 个独立 tanh-GELU 内核，带宽受限。此前 #28166 曾合并过无门控的 GELU epilogue 融合，但因数值偏差在 #28708 被回滚；#33453 引入的 quality 分层（lossless 默认 / high 可选）正好为这类非 bit-exact 优化提供了 opt-in 契约。该 PR 正是在这一机制下，在 lossless 路径提供 bit-exact 的残差门控融合，并在 high 路径提供 GELU epilogue。

实现拆解

1. 导入共享内核符号，新增 _flux_residual_gate_add 辅助函数（半精度 guard、torch.compiler.is_compiling() 逃逸、一次性 fallback），替换全部 5 个残差门控位点的 eager residual + gate * update。
2. 标记 3 类 tanh-GELU up-proj 位点：FluxGELU（TP>1 并行 FF）、新增 FluxFusedGELUProj 包装（TP=1 共享 diffusers 风格 FF，保留 net.0.proj 参数路径）、FluxSingleTransformerBlock.proj_mlp（单流非 nunchaku 分支）。位点默认 unmounted，由 DenoisingStage 对 quality=high 请求按批次 mount/unmount。
3. 门控与失败保护：can_fuse_linear_gelu 静态 + 动态 guard 拒绝无 bias、多 rank gather 输出、非半精度等场景；Nunchaku/ 量化路径从不标记，mount 的 all-or-nothing 检查也使其整体不生效。
4. 测试配套：在 test_fused_linear_gelu.py 新增 test_flux_gelu_proj_site，验证 unmount 默认位精确、mount 后近似、再 unmount 恢复；在 test_residual_gate_add.py 的 CASES 中追加 FLUX.1 三种真实形状 (1,4096,3072)、(1,512,3072)、(1,4608,3072) 位精确断言。

关键文件：

- python/sglang/multimodal_gen/runtime/models/dits/flux.py（模块 模型层；类别 source；类型 core-logic；符号 _flux_residual_gate_add, FluxFusedGELUProj, FluxGELU, FluxTransformerBlock）：核心接入文件：新增残差门控 helper、三个 GELU 融合位点及 FluxFusedGELUProj 包装，全部位点默认 unmounted，由质量层按请求挂载。
- test/registered/kernels/ops/diffusion/test_fused_linear_gelu.py（模块 融合内核；类别 test；类型 test-coverage；符号 test_flux_gelu_proj_site）：新增 FLUX 共享 FF 位点的 gate off/on 与 unmount 恢复 bit-exact 的测试，直接验证新包装类的行为。
- test/registered/kernels/ops/diffusion/test_residual_gate_add.py（模块 门控内核；类别 test；类型 test-coverage；符号 CASES）：为 residual-gate kernel 测试补充 FLUX.1 真实形状（4096/512/4608 x 3072），确保位精确断言覆盖实际推理 shape。

关键符号：_flux_residual_gate_add, FluxFusedGELUProj.forward, FluxGELU.forward, FluxSingleTransformerBlock.forward, test_flux_gelu_proj_site

关键源码片段

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

核心接入文件：新增残差门控 helper、三个 GELU 融合位点及 FluxFusedGELUProj 包装，全部位点默认 unmounted，由质量层按请求挂载。

```
# 定义于 python/sglang/multimodal_gen/runtime/models/dits/flux.py

# 全局开关：一旦 kernel 抛出异常，本进程内永久退回 eager 分支
_FLUX_RESIDUAL_GATE_CUDA_DISABLED = False


def _flux_residual_gate_add(
    residual: torch.Tensor,
    update: torch.Tensor,
    gate: torch.Tensor,
) -> torch.Tensor:
    """将 eager 两步 ``residual + gate * update`` 合并为单次 kernel 启动。

    仅在 fp16 / bf16 下使用：该精度下 kernel 与 eager 两步舍入逐位一致
    （``torch.equal`` 验证通过）；若为 fp32，融合会变成一次 fma（只舍入一次），
    破坏 bit-exact 语义，因此 fp32 保持原样。kernel 的行广播 gate 只支持
    ``[1, ..., 1, D]`` 形状，batch 维大于 1 时会触发 guard 失败并走 eager 分支。
    """
    global _FLUX_RESIDUAL_GATE_CUDA_DISABLED

    if (
        not _FLUX_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:
            # torch.compile 图内不允许静默 fallback：一旦编译进图，后续行为
            # 将与 eager 不一致，所以编译期直接抛出
            if torch.compiler.is_compiling():
                raise
            logger.warning_once(f"Disabling FLUX residual-gate CUDA fast path: {exc}")
            _FLUX_RESIDUAL_GATE_CUDA_DISABLED = True

    return residual + gate * update


class FluxFusedGELUProj(nn.Module):
    """TP=1 共享 FF 的 tanh-GELU up-proj 位点。

    取代 diffusers 的 ``GELU``（``approximate="tanh"``），但保留 ``net.0.proj``
    参数路径，因此 checkpoint 映射无需改动。默认 unmounted 时执行原始的
    Linear + tanh-GELU（bit-exact 参考路径）；DenoisingStage 对 quality="high"
    请求按批次 mount cublasLt epilogue 融合。
    """

    def __init__(self, proj: nn.Linear):
        super().__init__()
        self.proj = proj
        # 标记为融合位点，后续由 mount / unmount 协议统一控制
        mark_fused_gelu_site(self, "proj")

    def forward(self, hidden_states: torch.Tensor) -> torch.Tensor:
        if self._sgl_fused_gelu_enabled and can_fuse_linear_gelu(
            self.proj, hidden_states
        ):
            return fused_linear_gelu_tanh(
                hidden_states, self.proj.weight, self.proj.bias
            )
        return F.gelu(self.proj(hidden_states), approximate="tanh")
```

评论区精华

PR 没有任何 review 评论，唯一评论是 CI 链接。技术讨论主要体现在 PR 正文的设计权衡说明：GELU 融合必须限制在 quality=high，以此回应 #28708 回滚时“融合路径可能使生成图像偏离原始模型行为”的疑虑；残差门控则通过三层次位精确验证（kernel与 eagertorch.equal、推断 md5、全图 md5）从而可安全进入默认 lossless 路径。

- 暂无高价值评论线程

风险与影响

- 风险：

1. 只验证了 H200/SM90 环境，其他 GPU 上 residual_gate_add_cuda 可能因 guard 或 kernel 差异触发 fallback，好在 fallback 是原 *eager* 表达式，行为安全。
2. FluxFusedGELUProj 替换了 FluxTransformerBlock 或 FluxSingleTransformerBlock 中的 ff.net[0]，但保留 net.0.proj 参数路径，理论不破坏 checkpoint 加载；若其他代码硬编码了 GELU 类型或额外属性，存在隐性兼容风险。
3. 残差门控的 kernel 只支持 row-broadcast *gate* [1,...,1,D]，batch >1 时自动退回 *eager*，因此加速仅在 B=1 的场景生效；FLUX.1 服务通常单 batch，无实际影响。
4. torch.compile 图编译场景下 _flux_residual_gate_add 若遇到异常会直接抛出而非静默 fallback，避免编译图与 *eager* 不一致。 - 影响：用户侧：开启 quality=high 的 FLUX.1 请求获得约 4.3% 端到端加速，且经过 PSNR/SSIM 验证；默认 lossless 路径输出与 main 逐字节一致，不改变任何现有生成结果。系统侧：新增的位点协议和 helper 模式可复用于其他 DiT 模型，quality 分层机制得到又一次验证。团队侧：该 PR 为后续将非 bit-exact 融合安全落地提供了"默认精确 + 显式 opt-in"的模板。 - 风险标记：默认路径 bit-exact 保障，quality=high 门控，B>1 退回 *eager*, Nunchaku 路径不标记，torch.compile 逃逸处理

关联脉络

- 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): cublasLt GELU *epilogue* 内核与 mount/unmount 协议的来源，本 PR 在 FLUX.1 上接入该协议。
- PR #33734 [diffusion] ERNIE-Image bit-exact residual-gate fast path (H200 1024^2 *e2e* 16.17 -> 15.75 s): 同为 bit-exact residual-gate 接入，建立了 helper 形状、fallback 与验证协议，本 PR 沿用该模式。
- 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): quality=high 门控在 FLUX 系模型上的首个 fast path 先例，本 PR 进一步扩展该契约到 FLUX.1。