PR #34930 完整报告

sgl-project/sglang

[diffusion] Reuse bit-exact modulation fast path for LTX-2.3

合并时间：2026-08-17 09:04

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/34930

执行摘要

- 一句话：LTX-2.3 lossless 路径复用 bit-exact 调制快速路径
- 推荐动作：值得精读，尤其关注两个设计决策：其一是同一算术在 *eager* 与 *compile* 两种执行模式下选择不同实现（复用 kernel vs 保持表达式可见），这是多后端 / 多执行模式下 kernel 复用的典型案例；其二是用 monkeypatch 将 _ltx2_modulate 替换为 pytest.fail 来反向验证编译路径隔离，测试手法低成本高信号。建议阅读时结合 _ltx2_modulate 与 fused_ltx2_rms_norm_modulate 的差异，理解 "bit-exact 可复用、非 bit-exact 需挂载门控 " 的分级策略。

功能与动机

PR body 明确指出动机：lossless LTX-2.3 路径已经计算了 *reference* RMSNorm，随后又独立发起 broadcast scale/add 操作；共享的调制快速路径可以保留 RMSNorm 结果及其舍入，同时压缩两次大型 pointwise launch。作者还强调此优化独立于 LTX-2.3 启用 BCG（对应 #34929），说明这是单独的性能改进而非依赖 CUDA graph 改造。

实现拆解

实现分 4 步拆解：

1. 变更入口：python/sglang/multimodal_gen/runtime/models/dits/ltx_2.py 中的 _ltx2_rms_norm_modulate 函数，这是 LTX-2 所有 adaLN 位置的统一入口。
2. lossless 分支重构：原实现是 return rms_norm(x, eps) * (1 + scale) + shift 的 verbatim *eager* 链；改为先计算 normed = rms_norm(x, eps)，再按执行模式分叉：
   - torch.compiler.is_compiling() 为真时，保持 normed * (1 + scale) + shift 普通算术表达式，让 Inductor 将其融合进外层计算图；
   - *eager* 时调用 _ltx2_modulate(normed, scale, shift)，复用已被 quality=high 融合路径验证为 bit-exact 的调制 kernel，合并 broadcast scale 与 add 两次 launch。
3. 设计权衡：fused kernel（fused_ltx2_rms_norm_modulate）不是 bit-exact（误差 <= 1 bf16 ULP），故仍只由 quality=high 的请求级挂载控制；而 _ltx2_modulate 是 bit-exact 的，可安全用于 lossless 默认路径。编译期保持表达式可见是为了避免把 compiled call 路由进 opaque custom op，否则 Inductor 无法在图中做代数化简与 kernel 融合，构成回归。
4. 测试配套：test/registered/kernels/ops/diffusion/test_ltx2_rms_norm_modulate.py 新增 test_lossless_compile_keeps_expression_visible_to_inductor，用 monkeypatch 把

torch.compiler.is_compiling 置为 True，并将 _ltx2_modulate 替换为 pytest.fail，从反向证明编译路径不会调用不透明 custom op；同时更新 test_lossless_default_is_bitexact 的语义注释，继续以 torch.equal 严格断言 lossless 快速路径与 eager 表达式逐位一致。测试覆盖 video（hidden=4096）与 audio（hidden=2048）两种 hidden 尺寸。

关键文件：

- python/sglang/multimodal_gen/runtime/models/dits/ltx_2.py（模块 扩散模型；类别 source；类型 core-logic；符号 _ltx2_rms_norm_modulate）：核心变更文件：_ltx2_rms_norm_modulate 的 lossless 分支从 verbatim eager 表达式改为复用 _ltx2_modulate bit-exact 快速路径，并通过 torch.compiler.is_compiling() 分流保证编译期表达式对 Inductor 可见。
- test/registered/kernels/ops/diffusion/test_ltx2_rms_norm_modulate.py（模块 单元测试；类别 test；类型 test-coverage；符号 test_lossless_compile_keeps_expression_visible_to_inductor, test_lossless_default_is_bitexact）：测试配套：新增编译路径隔离测试，用 monkeypatch 反向验证 compile 分支永不调用 _ltx2_modulate，并强化 lossless bit-exact 断言语义；覆盖 video/audio 两种 hidden 尺寸。

关键符号：_ltx2_rms_norm_modulate, test_lossless_compile_keeps_expression_visible_to_inductor, test_lossless_default_is_bitexact

关键源码片段

python/sglang/multimodal_gen/runtime/models/dits/ltx_2.py

核心变更文件：_ltx2_rms_norm_modulate 的 lossless 分支从 verbatim eager 表达式改为复用 _ltx2_modulate bit-exact 快速路径，并通过 torch.compiler.is_compiling() 分流保证编译期表达式对 Inductor 可见。

```
def _ltx2_rms_norm_modulate(
    block: nn.Module,
    rms_norm: nn.Module,
    x: torch.Tensor,
    scale: torch.Tensor,
    shift: torch.Tensor,
    eps: float,
) -> torch.Tensor:
    """``rms_norm(x) * (1 + scale) + shift`` 作用于 LTX-2 的 adaLN 位置。

    当 ``quality="high"`` 融合已挂载到 ``block`` 且逐调用守卫通过时，
    将无权重 RMSNorm 与 modulate 折叠进单个 kernel；否则走 lossless
    默认路径。fused kernel 不是 bit-exact（误差 <= 1 bf16 ULP），
    因此只由请求级挂载控制，而不做运行时自检。
    """
    if ltx2_rms_norm_modulate_active(block) and can_fuse_ltx2_rms_norm_modulate(
        x, scale, shift
    ):
        # quality=high 挂载路径：走 fused kernel，允许 ULP 级误差
        return fused_ltx2_rms_norm_modulate(x, scale, shift, eps)
```

```
    # lossless 默认路径：先算出 reference RMSNorm 结果，保留其舍入
    normed = rms_norm(x, eps)

    if torch.compiler.is_compiling():
        # 编译期保留普通算术表达式，让 Inductor 融合进外层图；
        # 若路由到不透明的 custom op 会造成融合回归
        return normed * (1 + scale) + shift

    # eager 路径：复用已自验证 bit-exact 的调制快速路径，
    # 将 broadcast scale 与 add 两次 launch 折叠为一次 kernel 调用，
    # 且不改变 reference 舍入
    return _ltx2_modulate(normed, scale, shift)
```

test/registered/kernels/ops/diffusion/test_ltx2_rms_norm_modulate.py

测试配套：新增编译路径隔离测试，用 monkeypatch 反向验证 compile 分支永不调用 _ltx2_modulate，并强化 lossless bit-exact 断言语义；覆盖 video/audio 两种 hidden 尺寸。

```
def test_lossless_compile_keeps_expression_visible_to_inductor(monkeypatch):
    # 标记但未挂载的站点 = lossless 默认路径
    block = nn.Module()
    mark_ltx2_rms_norm_modulate_site(block)
    rms, x, scale, shift = _inputs(2048, seq=126)

    # 模拟编译期：此时表达式必须对 Inductor 可见
    monkeypatch.setattr(torch.compiler, "is_compiling", lambda: True)
    # 若编译路径调用了 opaque custom op，则直接让测试失败
    monkeypatch.setattr(
        ltx2_module,
        "_ltx2_modulate",
        lambda *_args: pytest.fail("compiled path must not call the opaque custom op"),
    )

    out = _ltx2_rms_norm_modulate(block, rms, x, scale, shift, 1e-6)
    # 编译分支走普通表达式，结果仍与 eager 引用逐位一致
    assert torch.equal(out, _eager(rms, x, scale, shift, 1e-6))
```

评论区精华

本 PR 无 review 评论（comments_count=0、review_comments_count=0），核心论证全部由作者 BBuf 在 PR body 中自证：

- 性能证据：引用 2026-08-14 全量 sweep（#21），重建后的 lossless 路径 one-stage eager denoise 为 24.019 s，torch.compile 为 24.005 s，两者在测量噪声内；two-stage presets 仍快于 compile。作者明确说明这是端到端 sweep 结果，非孤立微基准归因。
- bit-exact 依据：_ltx2_modulate 是 "first-sight-verified" 的 bit-exact eager kernel，复用不会改变 reference 舍入；而 fused kernel 因非 bit-exact 仍被限制在 quality=high 挂载路径。

- 独立性声明：作者强调本次优化与 LTX-2.3 启用 BCG 无关，避免与 #34929 的改动混淆。
  - 暂无高价值评论线程

风险与影响

- 风险：
  1. bit-exact 平台依赖：_ltx2_modulate 历史上曾因平台差异出现非 bit-exact 并回落 *eager*（见代码中 mismatch_msg="LTX-2 fused modulate is not bit-exact on this platform; falling back to *eager*" 的守卫逻辑）。lossless 默认路径直接复用它，若在未覆盖的硬件组合上触发非 bit-exact 分支，会静默破坏 lossless 契约；当前测试仅在 CUDA 上以 torch.equal 严格校验，AMD nightly 亦有覆盖，但其他后端需回归确认。
1. 编译 /*eager* 双路径一致性：torch.compiler.is_compiling() 是运行期判断，*capture* 阶段与执行阶段语义可能不一致；新增的 *compile* 路径测试用 monkeypatch 模拟了该分支，但未验证真实 torch.compile 捕获下 Inductor 的融合产物，存在理论上的图内数值差异风险。
2. 性能收益有限：B300 端到端数据中 lossless 重建路径与 *compile* 仅差 0.06%，收益落在测量噪声内；若目标用户的瓶颈不在这两个 broadcast launch，实际提速可能不可感知。
- 影响：
  - 用户影响：所有使用 LTX-2.3 quality=lossless（默认）的推理用户都会走到新路径，*denoise* 阶段减少两次大 broadcast pointwise launch，输出保持 bit-exact；quality=high 的挂载路径不受影响。
- 系统影响：改动仅两个文件、净增 27 行，局限在 multimodal_gen 的 LTX-2 模型目录内，不触及 SRT 调度、kernel 层或配置契约；_ltx2_rms_norm_modulate 是 adaLN 统一入口，所有调用点自动受益。
- 团队影响：为 diffusion lossless 路径确立了 "*eager* 复用 bit-exact kernel、*compile* 保持表达式可见 " 的双模式范式，后续 Ideogram、Sana 等模型的同类优化可复用该模式与测试手法。
- 风险标记：bit-exact 平台依赖，编译 /*eager* 双路径一致性，性能收益处于测量噪声内，缺少真实 torch.compile 端到端测试

关联脉络

- PR #34929 [diffusion] Enable breakable CUDA graphs for LTX-2.3: 同属 LTX-2.3 性能优化线，PR body 明确声明本次调制快速路径复用与 BCG 功能相互独立，避免耦合误判。
- PR #34928 [diffusion][kernel] Accelerate Sana BCG with bit-exact conv post-processing: 同为 diffusion 模型在 *eager* 路径上做 bit-exact 融合优化的方法论先例，共享 'bit-exact 可安全复用 ' 的设计原则。
- PR #34931 [diffusion] Accelerate lossless Ideogram norm post-processing: 同样针对 lossless 后处理路径复用共享内核以减少 launch，反映 multimodal_gen 中 lossless 路径 kernel 复用趋势。