PR #34315 完整报告

sgl-project/sglang

[diffusion] LTX-2: mount the bit-exact fused modulate at the 8 bare adaLN sites (ltx23-one-stage denoise -2.8% H100 / -2.6% H200)

合并时间：2026-08-11 18:23

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/34315

执行摘要

- 一句话：为 LTX-2 挂载位精确 CUDA 融合调制，denoise 最高降 2.8%
- 推荐动作：值得精读。该 PR 是“在位精确约束下安全挂载融合 kernel”的典型范例：BitExactFusionGate 首调自验证、(B,1,D) 步长视图的 densify 适配、以及分层回退策略都清晰可复用。结合 FLUX 与 MiniMax-H3 的同类挂载对比阅读，可沉淀一整套扩散模型 kernel 化性能优化的方法论。

功能与动机

LTX-2 block 中 8 处 x * (1 + scale) + shift adaLN 链未融合，每处需要 3 个 kernel（其中两个是对视频 / 音频流的全量遍历）。PR body 说明 bit-exact fused CUDA modulate（kernels/ops/diffusion/modulate_scale_shift.py，已被 FLUX 和 MiniMax-H3 挂载）可把每处折叠为 1 个 kernel，且按数值契约无需质量门控，从而在无质量损失的前提下缩短 denoise 时延。

实现拆解

1. 引入融合入口：在 python/sglang/multimodal_gen/runtime/models/dits/ltx_2.py 顶部新增 BitExactFusionGate 与 modulate_scale_shift 相关导入，并定义模块级 _LTX2_MODULATE 门与 _ltx2_modulate 助手函数。
2. 替换 8 个调用点：在 forward 中将 prompt 交叉注意力（视频 / 音频 2 处）、A2V/V2A 交叉注意力（4 处，分别作用于 video/audio 流）、输出层 norm_out 与 audio_norm_out（2 处）的裸表达式全部改为 _ltx2_modulate(...) 调用。
3. 布局适配：adaLN 行是来自 unbind/squeeze 的 (B,1,D) 步长视图，助手内部先 squeeze(1).contiguous() 密集成 kernel 期望的 (B,D) 连续布局；这只是一次微小的位精确拷贝。
4. 分层回退：per-token 调制行（dim 1 > 1）或平台不支持时保留 eager 链；首次融合调用通过 BitExactFusionGate.accept_or_fallback 与 eager 结果做 torch.equal 逐位比对，一致则后续直接走 kernel，不一致或抛异常则永久回退并记录日志。
5. 测试配套：新增 python/sglang/multimodal_gen/test/unit/test_ltx2_modulate_mount.py，覆盖 row-broadcast、strided unbind 视图与 eager 的位一致性，以及 per-token 行和 CPU 的回退分支；无配置、schema 或部署配套改动。

关键文件：

- python/sglang/multimodal_gen/runtime/models/dits/ltx_2.py（模块 LTX2 模型；类别 source；类型 core-logic；符号 _ltx2_modulate）：核心改动文件：新增 _ltx2_modulate 位精确融合助手，并将 forward 中 8 处裸 adaLN 调制统一替换为融合 kernel 调用，同时引入 BitExactFusionGate 自验证回退机制。
- python/sglang/multimodal_gen/test/unit/test_ltx2_modulate_mount.py（模块 单元测试；类别 test；类型 test-coverage；符号 TestLtx2ModulateMount, test_row_broadcast_matches_eager, test_non_contiguous_rows_match_eager, test_per_token_rows_fall_back）：新增单测文件，覆盖融合路径与 eager 链的位一致性，以及 per-token 行和 CPU 的回退分支，是正确性保障的关键配套。

关键符号：_ltx2_modulate, test_row_broadcast_matches_eager, test_non_contiguous_rows_match_eager, test_per_token_rows_fall_back, test_cpu_falls_back

关键源码片段

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

核心改动文件：新增 _ltx2_modulate 位精确融合助手，并将 forward 中 8 处裸 adaLN 调制统一替换为融合 kernel 调用，同时引入 BitExactFusionGate 自验证回退机制。

```
# `_ltx2_modulate`：把 8 处裸 `x * (1 + scale) + shift` 路由到位精确融合 kernel
# 关键点：
# - adaLN 的 scale/shift 通常是 `(B, 1, D)` 步长视图（来自 `unbind` / `squeeze`），
# 这里先 squeeze 并 contiguous 成 kernel 期望的 `(B, D)` 布局；
# - 只有第一次调用会做 `torch.equal` 对照 eager 链的位精确自验证，
# 通过后 `verified=True`，后续直接走 kernel；任何异常或不匹配都永久回退 eager。
_LTX2_MODULATE = BitExactFusionGate("LTX-2 fused modulate")


def _ltx2_modulate(
    x: torch.Tensor, scale: torch.Tensor, shift: torch.Tensor
) -> torch.Tensor:
    verified = _LTX2_MODULATE.verified
    # 只有满足 kernel 契约（3D、连续、行广播布局）且门未禁用时才尝试融合
    if (
        not _LTX2_MODULATE.disabled
        and x.dim() == 3
        and x.is_contiguous()
        and scale.dim() == 3
        and scale.shape == (x.shape[0], 1, x.shape[-1])
        and shift.shape == scale.shape
        and (verified or _LTX2_MODULATE.can_attempt_once())
    ):
        # 把 `(B, 1, D)` 的步长视图密集成 `(B, D)` 连续张量，位精确复制的开销极小
        scale_rows = scale.squeeze(1).contiguous()
        shift_rows = shift.squeeze(1).contiguous()
```

```
        if can_use_modulate_scale_shift_cuda(x, scale_rows, shift_rows):
            try:
                out = modulate_scale_shift_cuda(x, scale_rows, shift_rows)
            except Exception as exc:
                # kernel 抛出异常：记录并永久禁用融合路径
                _LTX2_MODULATE.on_exception(exc, logger=logger)
            else:
                if verified:
                    return out
                # 首次调用：与 eager 链逐位比对，一致才采纳，否则永久回退
                return _LTX2_MODULATE.accept_or_fallback(
                    out,
                    x * (1 + scale) + shift,
                    logger=logger,
                    mismatch_msg=(
                        "LTX-2 fused modulate is not bit-exact on this "
                        "platform; falling back to eager"
                    ),
                )
    # 不满足契约（如 per-token 行、CPU、非连续）一律走 eager 参考链
    return x * (1 + scale) + shift
```

python/sglang/multimodal_gen/test/unit/test_ltx2_modulate_mount.py

新增单测文件，覆盖融合路径与 eager 链的位一致性，以及 per-token 行和 CPU 的回退分支，是正确性保障的关键配套。

```
import unittest

import torch

from sglang.multimodal_gen.runtime.models.dits.ltx_2 import _ltx2_modulate


class TestLtx2ModulateMount(unittest.TestCase):
    # 常规 row-broadcast 布局：`(B, 1, D)` 的 scale/shift 应与 eager 链逐位一致
    @unittest.skipUnless(torch.cuda.is_available(), "requires CUDA")
    def test_row_broadcast_matches_eager(self):
        torch.manual_seed(0)
        x = torch.randn(2, 517, 4096, device="cuda", dtype=torch.bfloat16)
        scale = torch.randn(2, 1, 4096, device="cuda", dtype=torch.bfloat16)
        shift = torch.randn(2, 1, 4096, device="cuda", dtype=torch.bfloat16)
        reference = x * (1 + scale) + shift
        self.assertTrue(torch.equal(reference, _ltx2_modulate(x, scale, shift)))

    # `unbind()` / `squeeze()` 会产生 strided `(B, 1, D)` 视图，助手需先 densify
    @unittest.skipUnless(torch.cuda.is_available(), "requires CUDA")
    def test_non_contiguous_rows_match_eager(self):
        torch.manual_seed(1)
        x = torch.randn(2, 33, 512, device="cuda", dtype=torch.bfloat16)
```

```
        table = torch.randn(2, 1, 4, 512, device="cuda", dtype=torch.bfloat16)
        scale, shift = table.unbind(dim=2)[:2]
        reference = x * (1 + scale) + shift
        self.assertTrue(torch.equal(reference, _ltx2_modulate(x, scale, shift)))

    # per-token 行（dim 1 > 1）不满足 kernel 契约，应回退到 eager 链
    @unittest.skipUnless(torch.cuda.is_available(), "requires CUDA")
    def test_per_token_rows_fall_back(self):
        torch.manual_seed(2)
        x = torch.randn(2, 16, 128, device="cuda", dtype=torch.bfloat16)
        scale = torch.randn(2, 16, 128, device="cuda", dtype=torch.bfloat16)
        shift = torch.randn(2, 16, 128, device="cuda", dtype=torch.bfloat16)
        reference = x * (1 + scale) + shift
        self.assertTrue(torch.equal(reference, _ltx2_modulate(x, scale, shift)))

    # CPU 上无融合 kernel，必须回退到 eager 链
    def test_cpu_falls_back(self):
        torch.manual_seed(3)
        x = torch.randn(1, 9, 64, dtype=torch.float32)
        scale = torch.randn(1, 1, 64, dtype=torch.float32)
        shift = torch.randn(1, 1, 64, dtype=torch.float32)
        reference = x * (1 + scale) + shift
        self.assertTrue(torch.equal(reference, _ltx2_modulate(x, scale, shift)))


if __name__ == "__main__":
    unittest.main()
```

评论区精华

该 PR 无实质性 review 讨论，review_comments_count 与评论均为 0；唯一的一条 issue 评论是作者 BBuf 贴出的 CI 运行链接（Run #31421197616）。正确性论证主要依赖 PR body 中的性能对照表、ltx2 预设输出 md5 一致性、以及首调 torch.equal 自验证机制说明。

- 暂无高价值评论线程

风险与影响

- 风险：
  - 位精确契约依赖平台：自验证只在首次调用发生（can_attempt_once），若 kernel 在后续不同形状或批次上出现非位精确行为（低概率，因该 kernel 已在 FLUX/MiniMax-H3 使用），将无法再次拦截，只能依赖 eager 回退兜底。
  - 失败永久禁用：kernel 抛异常后进程内永久禁用融合路径，对长生命周期服务而言语义与 main 的 eager 链完全一致，但不会自动恢复。
  - ltx23-one-stage 缺乏端到端 md5 校验：该预设本身在未修改的 main 上就存在进程级不确定性，PR 只能依赖首调 torch.equal 门、确定性 ltx2 预设的一致性以及单测来保证正确性。

  - 性能回归风险：ltx2 预设 H100 上 6.582s 对 6.560s 属持平；首调 torch.equal 与 densify 拷贝的开销可以忽略，未见明显回归风险。
- 影响：
  - 用户侧：LTX-2 视频生成延迟降低，ltx23-one-stage denoise 时间 H100 上 -2.8%、H200 上 -2.6%；确定性 ltx2 预设输出 md5 不变，无质量回归。
  - 系统侧：每处融合从 3 个 kernel 降为 1 个，减少两次对视频 / 音频流的全量遍历，长视频 / 长时间生成收益更明显。
  - 团队侧：BitExactFusionGate 模式再次落地，连同 FLUX、MiniMax-H3、ERNIE/Ideogram 等形成可复用的"融合 kernel 安全挂载"模板；测试对布局契约与回退矩阵的覆盖方式值得在后续扩散优化中沿用。
  - 风险标记：核心生成路径变更，位精确契约依赖平台，首调用自验证后不再复核，ltx23 预设无端到端 md5 校验

关联脉络

- PR #34148 [MiniMax-H3] SubBlock: training-free block-sparse attention for the DiT: 同属 diffusion 推理性能优化线，且 PR body 明确提到 MiniMax-H3 已挂载同一 modulate_scale_shift kernel，本次是同一模式的又一落地。
- PR #34256 perf(vla): graph Pi0.5 prefix encoding: 同属 sglang/multimodal_gen 性能优化线，反映 diffusion/VLA 路径持续 kernel 化与 CUDA graph 化的整体演进方向。