PR #33546 完整报告

sgl-project/sglang

[diffusion] Wan VAE RMSNorm+SiLU fusion behind quality=high (H200 FastWan2.2 e2e 9.611 -> 9.125 s)

合并时间：2026-08-05 21:33

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/33546

执行摘要

- 一句话：Wan VAE 融合 kernel 置于 quality 门控，e2e 提速 5.1%
- 推荐动作：值得精读。核心看点：(1) 在『默认必须逐位一致』的硬约束下推进 kernel 融合的模式——quality 分层 + per-VAE 门控 + fail-closed 安装 + None 回退契约，是一套可复用的模板；(2) FQN 保持的 wrapper 设计（参数直接注册而非嵌套子模块），对任何『偷换模块』式优化都适用；(3) 与 torch.compile 的共存策略（is_compiling 让位 Inductor）。建议重点阅读 wan_vae_cuda_opt.py 的安装校验逻辑与 wan_rmsnorm_silu.py 的 dtype 边界处理。

功能与动机

PR body 明确说明这是复活 #30171（Fuse Wan VAE RMSNorm SiLU）：该 PR 因『consistency regression on B200』被关闭——融合 kernel 数值忠实（fp32 统计量、保持逐 eager 算子边界）但与 aten 非位级一致，会移动 CI golden 输出，属于策略约束而非 numerics bug。此后 #33453 将请求级 quality 限定为 lossless（默认）与 high 两档、#33451 提供 per-VAE 快路径门控与 DecodingStage 接线，解除了这一约束。工作量动机：FastWan2.2-TI2V-5B 是 3 步 DMD 视频模型，VAE 解码占请求 9.61 s 中的 5.77 s（60%），解码器中 29 处 WanRMS_norm→SiLU 链在最高 1.8 GB 激活上各自跑约 5 个 strided aten kernel。

实现拆解

1. 新增 Triton 融合 kernel（python/sglang/kernels/ops/diffusion/triton/wan_rmsnorm_silu.py，+196 行）

- 每个 program 处理一个 (b, t, h, w) 像素的通道行：channels_last_3d 布局下通道维最内层、加载完全合并；统计量用 fp32，按 WanRMS_norm.forward eager 的 dtype 边界逐步落盘，SiLU 在 fp32 计算。
- 相比 #30171 的三处刻意变更：(a) 支持 autocast 精度分裂（bf16 激活 + fp32 affine 参数，复现 aten 在 * gamma 处的 fp32 提升），否则 Wan2.1 bf16 解码永不触发融合；(b) 对不支持输入（非 CUDA、grad 模式、dtype 组合不支持、非 channels_last_3d、C > 1024）返回 None 而非抛异常，调用方必须 fallback；(c) t/h/w 尺寸改为运行时参数而非 tl.constexpr，避免因果分块 Wan 解码（t=1 头块 + t=2/4 稳态块）按分块形状反复重编译。

- 用 register_custom_op 登记 fake_impl，torch.compile 图捕获时只做 shape/stride 推导（输出沿用输入的 channels_last_3d stride）。

2. Wan 快路径封装与安装（python/sglang/multimodal_gen/runtime/models/vaes/wan_vae_cuda_opt.py，+141 行）

- FusedWanRMSNormSiLU 包装解码器中每条 WanRMS_norm→SiLU 链（残差块 norm1/norm2 + 输出头，FastWan2.2 VAE 共 29 处）：gate 开启且非编译时走融合 kernel（返回 None 则回退），否则执行与 norm + nn.SiLU 逐位一致的 F.silu(F.normalize(x, dim=1) * scale * gamma + bias)。
- gamma/bias 直接注册在 wrapper 上保持参数 FQN 不变（仍是 ...norm1.gamma），使按名权重迁移与严格 state_dict 加载不受影响——这是第 2 个 commit 修复组件精度 CI 后的最终形态。
- _install_norm_silu 采用全有或全无的 fail-closed 校验：任一残差块或输出头不是预期的 WanRMS_norm + 原生非 inplace nn.SiLU 结构即整体跳过并告警，避免部分融合造成行为分裂。
- maybe_optimize_wan_vae 只在 AutoencoderKLWan + WanDecoder3d、非空间并行解码（world_size > 1 跳过）、有 Triton 时安装；编码器完全不动（门控是 decode 作用域）；torch.compiler.is_compiling() 时主动让位给 Inductor（实测编译解码穿透自定义 op 反而更慢：4.90 → 4.98 s）。

3. 门控共享与平台接线

- VaeFastPathGate / GATE_ATTR 改为从 flux2_vae_cuda_opt 导入，避免两份门控类；flux2 文件仅压缩 docstring，无行为变化。
- python/sglang/multimodal_gen/runtime/platforms/cuda.py 的 optimize_vae 串接 maybe_optimize_wan_vae，异常时统一降级为未优化 VAE 并记日志。

4. 测试配套（test/registered/kernels/ops/diffusion/test_wan_vae_fastpath.py，+70 行）

- 用例 1：kernel 数值对比 eager 链，覆盖两种生产精度体制（FastWan fp32 统一 dtype；Wan2.1 bf16 x + fp32 affine、复现 aten fp32 提升），各带 / 不带 bias，并断言输出 dtype 与 stride（channels_last_3d）保持。
- 用例 2：wrapper 门控派发——gate off 时 torch.equal 与原始模块链严格一致、参数名保持 ['gamma']（按名权重迁移契约）；gate on 时路由到融合 kernel。
- 注册到 base-b-kernel-unit CUDA CI（1-gpu-large runner），跳过条件为无 CUDA。

关键文件：

- python/sglang/multimodal_gen/runtime/models/vaes/wan_vae_cuda_opt.py（模块 解码器优化；类别 source；类型 data-contract；符号 FusedWanRMSNormSiLU, init, forward, _is_plain_silu）：本 PR 的核心模块：FusedWanRMSNormSiLU wrapper 与 fail-closed 安装逻辑，决定 gate 开关、FQN 保持与逐位一致 off-path 行为，是 quality=high 门控模式在 Wan 家族的落地。

- python/sglang/kernels/ops/diffusion/triton/wan_rmsnorm_silu.py（模块 融合内核；类别 infra；类型 infrastructure；符号 _wan_rmsnorm_silu_kernel, _fake_wan_rmsnorm_silu, _triton_wan_rmsnorm_silu_cuda, _affine_supported）：性能收益的载体：channels_last_3d 单 kernel 融合（最高 4.48x 微观加速），其 fp32 统计量、dtype 边界复现、None 回退契约与运行时 shape 参数设计是数值与编译行为的关键。
- test/registered/kernels/ops/diffusion/test_wan_vae_fastpath.py（模块 单元测试；类别 test；类型 test-coverage；符号 _cl3d, test_kernel_numerics, test_fused_module_gate_dispatch）：验证 kernel 在两种生产精度体制下的数值正确性、输出布局保持，以及 gate off 时 wrapper 与原始模块链的逐位一致和参数 FQN 契约。
- python/sglang/multimodal_gen/runtime/models/vaes/flux2_vae_cuda_opt.py（模块 解码器优化；类别 source；类型 data-contract；符号 VaeFastPathGate, GATE_ATTR, FusedGroupNormSiLU）：VaeFastPathGate/GATE_ATTR 的共享来源：本 PR 从此处导入门控类并压缩 docstring，保证两个 VAE 快路径模块共用同一契约且保持行为不变。
- python/sglang/multimodal_gen/runtime/platforms/cuda.py（模块 平台接线；类别 source；类型 dependency-wiring；符号 CudaPlatformBase.optimize_vae）：把 Wan VAE 快路径接入现有 optimize_vae 平台钩子：maybe_optimize_wan_vae 与 maybe_optimize_flux2_vae 串接，异常统一降级为未优化 VAE。

关键符号：maybe_optimize_wan_vae, _install_norm_silu, FusedWanRMSNormSiLU.forward, wan_rmsnorm_silu, _triton_wan_rmsnorm_silu_cuda, can_use_wan_rmsnorm_silu

关键源码片段

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

验证 kernel 在两种生产精度体制下的数值正确性、输出布局保持，以及 gate off 时 wrapper 与原始模块链的逐位一致和参数 FQN 契约。

```
'''Wan VAE 解码器快路径测试：融合 kernel 数值与门控派发（lossless off-path 必须逐位一致）。'''

import pytest
import torch
import torch.nn as nn
import torch.nn.functional as F

from sglang.kernels.ops.diffusion.triton.wan_rmsnorm_silu import wan_rmsnorm_silu
from sglang.multimodal_gen.runtime.models.vaes.wan_vae_cuda_opt import (
    FusedWanRMSNormSiLU,
    VaeFastPathGate,
)
from sglang.multimodal_gen.runtime.models.vaes.wanvae import WanRMS_norm
from sglang.test.ci.ci_register import register_cuda_ci

register_cuda_ci(est_time=40, stage='base-b-kernel-unit', runner_config='1-gpu-large')

pytestmark = pytest.mark.skipif(not torch.cuda.is_available(), reason='CUDA required')
```

```
def _cl3d(shape, dtype):
    # 生成 channels_last_3d 连续张量，模拟 VAE 解码的真实布局
    return torch.randn(shape, device='cuda', dtype=dtype).contiguous(
        memory_format=torch.channels_last_3d
    )


@torch.no_grad()
@pytest.mark.parametrize(
    'x_dtype,affine_dtype,atol,rtol',
    [
        (torch.float32, torch.float32, 1e-5, 1e-5), # FastWan2.2 fp32 解码
        (torch.bfloat16, torch.float32, 1.5e-1, 3e-2), # Wan2.1 bf16 autocast
    ],
)
def test_kernel_numerics(x_dtype, affine_dtype, atol, rtol) -> None:
    torch.cuda.manual_seed(0)
    x = _cl3d((1, 96, 3, 10, 14), x_dtype)
    gamma = torch.randn((96, 1, 1, 1), device='cuda', dtype=affine_dtype)
    for bias in (None, torch.randn_like(gamma)):
        # eager 参考链：F.normalize(x, dim=1) * scale * gamma (+bias) + SiLU
        expected = F.silu(
            F.normalize(x, dim=1) * 96**0.5 * gamma + (0 if bias is None else bias)
        )
        actual = wan_rmsnorm_silu(x, gamma, bias)
        assert actual is not None and actual.dtype == expected.dtype
        # 输出必须保持输入 stride（VAE 依赖 channels_last_3d 布局）
        assert actual.stride() == x.stride()
        torch.testing.assert_close(actual, expected, atol=atol, rtol=rtol)


@torch.no_grad()
def test_fused_module_gate_dispatch() -> None:
    # gate off 必须逐位一致；gate on 必须路由到融合 kernel
    torch.cuda.manual_seed(0)
    norm = WanRMS_norm(96, images=False).to(device='cuda', dtype=torch.bfloat16)
    norm.gamma.add_(torch.randn_like(norm.gamma))
    gate = VaeFastPathGate()
    fused = FusedWanRMSNormSiLU(norm, gate)
    # 参数名必须保持不变（按名权重迁移的契约）
    assert [n for n, _ in fused.named_parameters()] == ['gamma']
    x = _cl3d((1, 96, 3, 10, 14), torch.bfloat16)
    assert torch.equal(fused(x), nn.SiLU()(norm(x)))
    gate.enabled = True
    expected = wan_rmsnorm_silu(x, norm.gamma, rms_scale=float(norm.scale))
    assert torch.equal(fused(x), expected)


if __name__ == '__main__':
```

```
import sys

sys.exit(pytest.main([__file__, '-v', '-s']))
```

评论区精华

该 PR 没有 review 评论线程，争议与决策主要沉淀在 PR body 和 issue 评论中：

1. 一致性回归策略：原 #30171 因融合 kernel 与 aten 非位级一致、移动了 CI golden 输出而被关闭。本 PR 明确这是『策略约束而非 numerics bug』，并借助 #33453（quality 限定 lossless/high 两档）+ #33451（per-VAE 门控 + DecodingStage 接线）把 kernel 收敛到显式 opt-in 的 quality=high，默认路径 sha256 逐字节不变。
2. 组件精度 CI 失败与 FQN 修复：BBuf 在评论中记录了 wrapper 曾把整个 WanRMS_norm 嵌套为子模块，导致 ...norm1.gamma → ...norm1.norm.gamma 改名（29 个参数），按名权重迁移检查失败（81/110、167/196 < 98%）；修复为直接注册 gamma/bias、内联逐位一致的 off-path 链，并用 named_parameters 对比 + 全 WanDecoder3d 严格 state_dict 回环验证。
3. torch.compile 取舍：编译解码路由自定义 op 反而更慢（4.90 → 4.98 s），故 is_compiling() 时走原始路径，两档 quality 在编译模式下均保持 parity。

- B200 一致性回归与 quality 门控策略 (design): 采用质量门控方案：默认 lossless 路径 sha256 逐字节不变，golden CI 不受影响；仅 quality=high 请求运行融合 kernel。
- 组件精度 CI 失败：wrapper 参数 FQN 保持 (correctness): gamma/bias 直接注册，FQN 不变；named_parameters 安装前后一致 + 全 WanDecoder3d 严格 state_dict 回环验证通过；VaeFastPathGate 改为从 flux2_vae_cuda_opt 导入去重。
- torch.compile 下自定义 op 与 Inductor 的取舍 (design): is_compiling() 时走原始模块路径，两档 quality 在编译模式下均保持 parity（4.90 s flat）。

风险与影响

- 风险：
  - 数值非位级一致仅限 quality=high：融合路径与 aten 在 fp32 FastWan 解码上偏差 mean abs 1.1e-5、p99 7.9e-5，bf16 autocast 路径偏差与 baseline 统计等价（mean abs 0.001003 vs 0.000959），已用 PSNR > 25 dB 与逐帧 SSIM 把关；但依赖 reduction 顺序的个别像素仍可能超出测试容差，对逐帧 bit 级敏感的应用需避免使用 quality=high。
  - 默认逐位保证依赖结构匹配检查：_install_norm_silu 对 WanResidualBlock / WanRMS_norm 做 strict type 与属性检查（channel_first、Tensor gamma、非 inplace nn.SiLU）。一旦未来 Wan VAE 结构变化（继承、包装、新增激活），要么 fail-closed 静默降级（安全方向），要么因 type 严格判断误跳过优化（性能方向），两种都值得注意。
  - 多步模型收益稀释：50 步请求中同一 0.66 s 解码增量被稀释到约 1-2%，收益感知与 workload 强相关。

  - kernel 支持面有限：仅 CUDA、无梯度、5D channels_last_3d、C <= 1024、限定 dtype；且 Wan2.1 的 50 步 e2e A/B 因 UMT5 多分片权重映射回归暂时无法复现（body 已说明，与本次改动无关）。
  - 回归面：默认路径无 kernel 参与，golden CI 不受影响；编译解码两档 parity；空间并行解码与无 Triton 环境自动跳过，风险可控。
- 影响：
  - 用户：--quality high 显式开启后才走融合 kernel，FastWan2.2-TI2V-5B 720p×81 帧 e2e -5.1%（9.611 → 9.125 s），解码阶段 -7.6%；Wan2.1 480p 解码 -11.9%；默认 lossless 与 main 逐字节一致，CI golden 无扰动。
  - 系统：每 VAE 加载时一次性结构校验与安装（29 处包装），运行期仅一个 enabled 标志判断；kernel 按实际分块形状在首次调用时编译（t/h/w 运行时参数避免了因果解码的重复重编译）。
  - 团队 / 生态：确立『非位级一致优化一律放 quality=high 门控』的开发范式，VAE 快路径从 FLUX.2 扩展到 Wan 家族；VaeFastPathGate 去重后成为两个 VAE 模块共享的契约，后续新融合可直接复用。
  - 范围：全部改动限定在 multimodal_gen 运行时与 diffusion kernel 目录，不触碰 LLM serving 主路径。
  - 风险标记：数值非位级一致仅限 quality=high，默认逐位保证依赖结构匹配检查，多步模型收益稀释至 1-2%，kernel 仅覆盖 CUDA 限定 dtype，bf16 数值验证为统计等价

关联脉络

- PR #30171 [diffusion] Fuse Wan VAE RMSNorm SiLU: 本 PR 直接复活其 kernel 与工作量动机；该 PR 因 B200 consistency regression（golden 输出被移动）关闭，本 PR 用 quality=high 门控解除了这一策略约束。
- PR #33451 Add per-VAE fast-path gate and DecodingStage wiring: 提供 VaeFastPathGate 与 DecodingStage 的 _sgl_vae_fast_path_gate 接线（duck-typing），本 PR 原样复用其门控机制，DecodingStage 零改动。
- PR #33453 Restrict request-level quality to lossless/high tiers: 将请求级 quality 限定为 lossless（默认）与 high 两档，为把非位级一致的融合 kernel 收敛到显式 opt-in 提供了前提条件。