PR #35728 完整报告

sgl-project/sglang

[diffusion] Accelerate SANA-Video linear attention in quality=high

合并时间：2026-08-21 18:05

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/35728

执行摘要

- 一句话：SANA-Video high 档线性注意力新增 BF16 快路径，提速约 5%
- 推荐动作：值得精读。核心看点：try_sana_video_linear_attention 的多条件 guard 与回退设计、QualityGatedFusion 的请求级挂载复用、以及如何在保持默认路径逐位一致的前提下为单个质量档开放 BF16 加速。建议顺带检查 torch.bmm(out_dtype=...) 的最低版本要求，并考虑把模型级 A/B 验证固化到 CI。

功能与动机

PR body 给出了明确的提速诉求与数据：SANA-Video 在 quality=high 下默认 eager 链会在两次线性注意力 GEMM 前把 Q/K/V 提升到 FP32，开销较大。作者让第一个 GEMM 保持 BF16 输入并请求 FP32 累加 / 输出，第二个 GEMM 仍走 FP32，H200 denoise 阶段从 243.8/244.7 ms/step 降到 231.7/232.6 ms/step（约 5.0%），同时声明 quality=lossless 路径逐位不变。核心动机是在不动默认行为的前提下，为 high 质量档利用 BF16 Tensor Core 提速。

实现拆解

1. 门控站点基建：新增 python/sglang/kernels/ops/diffusion/sites/sana_video_linear_attention_site.py，基于 QualityGatedFusion 定义 mark/mount/unmount/active/try 五个函数；try_sana_video_linear_attention 对 CUDA、BF16、4 维布局、batch=1 等条件做 guard，不满足返回 None 并回退参考链。
2. 模型接入：sana_video.py 的 SanaVideoLinearAttention.__init__ 中 mark_sana_video_linear_attention_site(self)；forward 移除强制 .float()，改为先尝试 BF16 快路径，失败回退 FP32 链。
3. 去噪管线挂载：denoising.py 的 _QUALITY_FUSION_HANDLERS 注册 mount/unmount_sana_video_linear_attention，让 quality=high 请求在批次边界挂载 / 卸载。
4. 导出与文档：diffusion/__init__.py 懒注册表登记 5 个符号；fused_kernels.mdx、README.md、benchmark skill 文档同步更新。
5. 测试配套：test_sites.py 新增 CUDA 单测覆盖挂载生命周期与数值容差；模型级 SSIM/PSNR 与 SHA256 验证在 H200/B300 手动执行。

关键文件：

- python/sglang/kernels/ops/diffusion/sites/sana_video_linear_attention_site.py（模块 门控站点；类别 infra；类型 infrastructure；符号 mark_sana_video_linear_attention_site, sana_video_linear_attention_active, _site_reject_reason, mount_sana_video_linear_attention）：新增质量门控站点，封装 BF16 输入线性注意力的标记、挂载、卸载、尝试执行与状态查询，是本次加速的核心基础设施。
- python/sglang/multimodal_gen/runtime/models/dits/sana_video.py（模块 模型实现；类别 source；类型 core-logic；符号 SanaVideoLinearAttention, SanaVideoLinearAttention.forward）：模型侧在 SanaVideoLinearAttention 上标记站点并改写 forward，将强制 FP32 链改为先尝试 BF16 快路径、失败回退 FP32，是本行为的直接入口。
- test/registered/kernels/ops/diffusion/test_sites.py（模块 单元测试；类别 test；类型 test-coverage；符号 test_sana_video_linear_attention_quality_path_and_guards）：新增针对该快路径的 CUDA 单测，覆盖未挂载拒绝、挂载后数值与 FP32 参考接近、非单 batch 拒绝、卸载恢复。
- python/sglang/multimodal_gen/runtime/pipelines_core/stages/denoising.py（模块 去噪管线；类别 source；类型 core-logic；符号 _QUALITY_FUSION_HANDLERS）：把 mount/unmount 注册进 _QUALITY_FUSION_HANDLERS，使 quality=high 请求在去噪循环中统一挂载卸载。
- python/sglang/kernels/ops/diffusion/__init__.py（模块 符号导出；类别 infra；类型 infrastructure；符号 mark_sana_video_linear_attention_site, mount_sana_video_linear_attention, sana_video_linear_attention_active, try_sana_video_linear_attention）：在延迟符号注册表中登记 5 个新公开符号，保证 from sglang.kernels.ops.diffusion import ... 可用。
- docs/docs/sglang-diffusion/fused_kernels.mdx（模块 文档；类别 other；类型 documentation）：在 fusion families 表中加入 SANA-Video BF16 线性注意力条目，并更新 kernel inventory。
- python/sglang/kernels/ops/diffusion/README.md（模块 文档；类别 docs；类型 documentation）：补充 SANA-Video 线性注意力条目到支持模型列表。
- python/sglang/multimodal_gen/.claude/skills/sglang-diffusion-benchmark-profile/SKILL.md（模块 基准技能；类别 docs；类型 documentation）：更新 benchmark skill 指引，说明 SANA-Video high 质量快路径。
- python/sglang/multimodal_gen/.claude/skills/sglang-diffusion-benchmark-profile/benchmark-and-profile.md（模块 基准技能；类别 docs；类型 documentation）：调整 benchmark 文档中的一处表述。

关键符号：mark_sana_video_linear_attention_site, sana_video_linear_attention_active, mount_sana_video_linear_attention, unmount_sana_video_linear_attention, try_sana_video_linear_attention, SanaVideoLinearAttention.forward, test_sana_video_linear_attention_quality_path_and_guards

关键源码片段

python/sglang/kernels/ops/diffusion/sites/sana_video_linear_attention_site.py

新增质量门控站点，封装 BF16 输入线性注意力的标记、挂载、卸载、尝试执行与状态查询，是本次加速的核心基础设施。

```
# SANA-Video BF16-input linear attention，按请求 quality 门控。
# 参考实现把旋转后的 Q/K 与 V 在两次 attention GEMM 前都提升到 FP32；
# 这里在 quality=high 时让第一个 GEMM 保持 BF16 输入、FP32 累加 / 输出，
# 第二个 GEMM 仍走 FP32。默认 quality=lossless 保持原 FP32 输入链逐位一致。
from __future__ import annotations

import logging

import torch
import torch.nn as nn

from sglang.kernels.ops.diffusion.sites.quality_gate import QualityGatedFusion

logger = logging.getLogger(__name__)

_FUSION = QualityGatedFusion(
    name='SANA-Video BF16-input linear attention',
    marker_attr='_sgl_sana_video_linear_attention_site',
    enabled_attr='_sgl_sana_video_linear_attention_enabled',
)


# 标记站点：让 QualityGatedFusion 能识别 SANA-Video 的线性注意力模块。
def mark_sana_video_linear_attention_site(module: nn.Module) -> None:
    _FUSION.mark(module)


# 查询当前请求是否已挂载 BF16 输入路径。
def sana_video_linear_attention_active(module: nn.Module) -> bool:
    return _FUSION.is_enabled(module)


# 站点级拒绝理由：非 CUDA 环境直接拒绝，避免误用。
def _site_reject_reason(_site: nn.Module) -> str | None:
    if torch.version.cuda is None:
        return 'CUDA is unavailable'
    return None


# 挂载快路径；内部只对带标记属性的站点生效，全或无。
def mount_sana_video_linear_attention(root: nn.Module) -> bool:
    return _FUSION.mount(root, reject_reason=_site_reject_reason, logger=logger)

# 卸载快路径，恢复参考链；卸载后应逐位一致。
```

```
def unmount_sana_video_linear_attention(root: nn.Module) -> None:
    _FUSION.unmount(root)


# 核心入口：返回质量门控下的注意力结果，条件不满足时返回 None 让调用方回退 FP32 链。
def try_sana_video_linear_attention(
    site: nn.Module,
    query_rotate: torch.Tensor,
    key_rotate: torch.Tensor,
    value: torch.Tensor,
    normalizer: torch.Tensor,
) -> torch.Tensor | None:
    # guards：站点已启用、CUDA、BF16 且三者 dtype 一致、4 维、
    # 三者 shape 相等、batch size 为 1、normalizer 在 CUDA 上。
    if not (
        _FUSION.is_enabled(site)
        and query_rotate.is_cuda
        and query_rotate.dtype == torch.bfloat16
        and key_rotate.dtype == query_rotate.dtype
        and value.dtype == query_rotate.dtype
        and query_rotate.dim() == 4
        and query_rotate.shape == key_rotate.shape == value.shape
        and query_rotate.shape[0] == 1
        and normalizer.is_cuda
    ):
        return None

    # 第一个 GEMM：BF16 输入 + FP32 输出 / 累加，利用 BF16 Tensor Core 提速。
    batch_size, num_heads, head_dim, _ = value.shape
    scores = torch.bmm(
        value.flatten(0, 1),
        key_rotate.transpose(-1, -2).flatten(0, 1),
        out_dtype=torch.float32,
    ).view(batch_size, num_heads, head_dim, head_dim)

    # 第二个 GEMM：保持 FP32，与参考链的数值语义一致。
    return (scores @ query_rotate.float()) * normalizer
```

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

模型侧在 SanaVideoLinearAttention 上标记站点并改写 forward，将强制 FP32 链改为先尝试 BF16 快路径、失败回退 FP32，是本行为的直接入口。

```
def forward(
    self,
    hidden_states: torch.Tensor,
    rotary_emb: tuple[torch.Tensor, torch.Tensor],
) -> torch.Tensor:
    original_dtype = hidden_states.dtype
    batch_size, sequence_length, _ = hidden_states.shape
```

```
    # 投影出 packed QKV 并按头拆分。
    qkv, _ = self.to_qkv(hidden_states)
    query, key, value = qkv.split(self.inner_dim, dim=-1)

    # 分别对 Q/K 做 RMSNorm 后 reshape 为 [B, N, H, D]。
    query = self.norm_q(query).view(
        batch_size, sequence_length, self.num_heads, self.head_dim
    )
    key = self.norm_k(key).view(
        batch_size, sequence_length, self.num_heads, self.head_dim
    )
    value = value.view(batch_size, sequence_length, self.num_heads, self.head_dim)

    # ReLU 激活 + interleaved RoPE。
    query = F.relu(query)
    key = F.relu(key)
    query_rotate = apply_interleaved_rotary_emb(query, *rotary_emb)
    key_rotate = apply_interleaved_rotary_emb(key, *rotary_emb)

    # 转置到 [B, H, D, N] 布局。
    query = query.permute(0, 2, 3, 1)
    key = key.permute(0, 2, 3, 1)
    query_rotate = query_rotate.permute(0, 2, 3, 1)
    key_rotate = key_rotate.permute(0, 2, 3, 1)
    value = value.permute(0, 2, 3, 1)

    # 归一化分母在原始 dtype 上计算，与参考路径保持一致。
    normalizer = 1.0 / (
        key.sum(dim=-1, keepdim=True).transpose(-2, -1) @ query + 1e-15
    )

    # 优先走 BF16 输入快路径；站点未挂载或布局不满足时返回 None，回退 FP32 链。
    hidden_states = try_sana_video_linear_attention(
        self, query_rotate, key_rotate, value, normalizer
    )
    if hidden_states is None:
        scores = value.float() @ key_rotate.float().transpose(-1, -2)
        hidden_states = (scores @ query_rotate.float()) * normalizer

    hidden_states = hidden_states.flatten(1, 2).transpose(1, 2)
    hidden_states = hidden_states.to(original_dtype)
    return self.to_out[0](hidden_states)
```

评论区精华

本 PR 无 review 评论。评论区仅有作者 BBuf 的两条 /rerun-failed-ci 和一条 workflow 链接；PR Test (Extra) 曾失败，Base 与 AMD ROCm 7.2 通道等待重跑。核心设计决策（BF16 输入配合 FP32 累加、lossless 位级一致）仅在 PR body 中自证，没有 reviewer 提出质疑。

- PR CI 状态与重跑 (other): 没有 reviewer 参与；作者重跑 CI 后合并，设计权衡由 PR body 自证。

风险与影响

- 风险：
  - 数值精度：high 档输出与 FP32 参考存在约 1e-2 差异，仅靠 body 中 SSIM/PSNR 与 A/B 视频验证，缺少多 seed/ 分辨率统计。
  - 适用面：快路径严格限定 CUDA、BF16、单 batch、[B=1, H, D, N] 布局，其余自动回退 FP32，正确性有保障但批量或非 CUDA 无收益。
  - 版本依赖：torch.bmm 的 out_dtype 参数需要较新 PyTorch，若环境过旧可能在快路径抛错而非回退（材料未确认，需验证）。
  - 门控机制：新增 handler 进入 _QUALITY_FUSION_HANDLERS，若与其他 quality-gated fusion 交互异常可能残留状态，但该模式已在多个 diffusion 模型上复用。
  - 测试覆盖：模型级视频质量验证未纳入自动 CI，仅靠手动 H200/B300 结果，存在回归不被及时发现的风险。
  - 影响：影响范围集中在 SANA-Video 的 quality=high 场景：用户可获约 5% 至 7.6% 的 denoise/e2e 提速，峰值显存不变；quality=lossless 输出位级一致，其他 diffusion 模型不受影响。对团队而言，新增了一个可复用的 QualityGatedFusion 站点模式，未来可在其他模型的融合 kernel 上直接套用，并需要维护与参考链的一致性回归。
  - 风险标记：high 档数值精度依赖 A/B 验证，仅 CUDA 单 batch 生效，torch.bmm out_dtype 版本依赖，模型级测试未入 CI，新增质量门控注册

关联脉络

- PR #35698 [diffusion] Fuse LTX-2.5 decoder 3D RoPE: 同属 diffusion fused-kernel 演进线，同样在 python/sglang/kernels/ops/diffusion/init.py 与 docs/docs/sglang-diffusion/fused_kernels.mdx 中登记新融合 kernel 并配套单测 /bench。
- PR #35701 [diffusion] feat: let offloaded weights stay on the checkpoint mapping: 同属 python/sglang/multimodal_gen/runtime 的 diffusion 运行路径，反映 diffusion 子系统的持续演进与共享内存 / 加载基础设施变化。