PR #25985 完整报告

sgl-project/sglang

[diffusion][bugfix] Fix Wan channels_last_3d VAE decode corruption

合并时间：2026-05-22 23:34

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/25985

执行摘要

- 一句话：修复 Wan VAE 解码 channels_last_3d 格式导致的视觉损坏
- 推荐动作：值得所有涉及扩散模型推理的开发者精读，尤其是处理内存格式兼容性和分布式通信的注意事项。该 PR 展示了如何平衡性能与正确性。

功能与动机

当 Conv3d 权重以 channels_last_3d 格式存储时，输入张量默认是 contiguous 格式，PyTorch 的 Conv3d 运算可能产生未定义结果（视觉损坏）。该问题在 8 卡 Wan2.2 I2V 服务中复现，生成视频出现视觉混乱。

实现拆解

1. 在 wan_common_utils.py 中新增 _conv3d_weight_is_channels_last_3d 检测函数和 match_conv3d_input_format 匹配函数，并在 WanCausalConv3d.forward 中调用，确保输入格式与权重一致。
2. 在 wan_dist_utils.py 中新增 _maybe_contiguous_for_sp_gather 函数，用于在 all_gather 前将 channels_last_3d 张量转换为 contiguous；在 gather_and_trim_height 和 WanDistCausalConv3d.forward 中调用，避免分布式通信错误。
3. 更新测试数据基线 SGL_TEST_FILES_CI_DATA_REVISION 以匹配新输出。

关键文件：

- python/sglang/multimodal_gen/runtime/models/vaes/parallel/wan_common_utils.py（模块 扩散模型；类别 source；类型 core-logic；符号 _conv3d_weight_is_channels_last_3d, match_conv3d_input_format）：核心修复文件：新增格式检测与匹配函数，并修改 WanCausalConv3d.forward 调用，确保 Conv3d 输入格式与权重一致。
- python/sglang/multimodal_gen/runtime/models/vaes/parallel/wan_dist_utils.py（模块 扩散模型；类别 source；类型 core-logic；符号 _maybe_contiguous_for_sp_gather）：分布式兼容修复：新增 _maybe_contiguous_for_sp_gather 函数，在 all_gather 前转换非 contiguous 张量；引入 match_conv3d_input_format 并在 WanDistCausalConv3d.forward 中调用。
- python/sglang/multimodal_gen/test/test_utils.py（模块 测试工具；类别 test；类型 test-coverage）：更新 CI 数据基线 revision 以匹配合并后的新输出，确保准确性测试通过。

关键符号：_conv3d_weight_is_channels_last_3d, match_conv3d_input_format, _maybe_contiguous_for_sp_gather

关键源码片段

python/sglang/multimodal_gen/runtime/models/vaes/parallel/wan_common_utils.py

核心修复文件：新增格式检测与匹配函数，并修改 WanCausalConv3d.forward 调用，确保 Conv3d 输入格式与权重一致。

```
def _conv3d_weight_is_channels_last_3d(weight: torch.Tensor) -> bool:
    # 检测 Conv3d 权重是否以 channels_last_3d 格式存储
    return (
        weight.dim() == 5
        and hasattr(torch, "channels_last_3d")
        and weight.is_contiguous(memory_format=torch.channels_last_3d)
    )


def match_conv3d_input_format(x: torch.Tensor, weight: torch.Tensor) -> torch.Tensor:
    # 如果权重是 channels_last_3d，则将输入也转为该格式，否则保持原样
    if x.dim() == 5 and _conv3d_weight_is_channels_last_3d(weight):
        return x.contiguous(memory_format=torch.channels_last_3d)
    return x


# 在 WanCausalConv3d.forward 中使用：
def forward(self, x, cache_x=None):
    # ... 其他处理 ...
    x = F.pad(x, padding)
    x = (
        x if current_platform.is_amp_supported() else x.to(self.weight.dtype)
    )
    x = match_conv3d_input_format(x, self.weight) # 确保输入格式与权重一致
    return super().forward(x)
```

评论区精华

mickqian 注意到初始方案在每个 Conv3d 输出处都转回 contiguous 会导致输入再次转换，带来额外开销。最终优化为只在输入处匹配权重格式，并在 all_gather 前执行一次 contiguous，显著降低开销。同时需要更新准确性基线以匹配修复后的输出。

- 初始修复的性能开销分析 (performance): 优化方案被接受，PR 合并。
- 准确性基线更新 (testing): 作者更新了 SGL_TEST_FILES_CI_DATA_REVISION，新基线已设置。

风险与影响

- 风险：
  1. 依赖 PyTorch 的 channels_last_3d 支持，老版本可能不存在该格式。代码中通过 hasattr 检查，兼容性较好。
  2. 修改了 VAE 解码的关键路径，可能影响所有 Wan 模型，但已有 CI 和 GT 验证。
  3. 在 all_gather 前增加 contiguous 调用，可能带来微小开销，但避免了数据损坏。 - 影响：对用户：修复了高分辨率视频生成任务中的视觉损坏。 对系统：提升了内存格式处理的健壮性。 对团队：需要维护新增的格式检测和匹配逻辑，并确保未来 Conv3d 相关改动考虑格式兼容。
- 风险标记：依赖 channels_last_3d 的 PyTorch 版本，需要更新准确性基线，分布式通信前新增格式转换

关联脉络

- PR #25674 [diffusion] Fix MOVA DAC bf16 on ROCm: 同为 diffusion 模块的 bugfix，涉及 VAE 或解码兼容性，与本 PR 在同一目录。
- PR #25168 [diffusion] Role-based component loading and stage affinity: 扩散模型架构调整，涉及同一个运行时核心代码目录。