PR #26121 完整报告

sgl-project/sglang

[diffusion] Auto-select VAE channels_last_3d

合并时间：2026-05-23 10:20

原文链接：http://prhub.com.cn/sgl-project/sglang/pull/26121

执行摘要

本 PR 为 VAE 的 Conv3d weights 引入自动选择 channels_last_3d 内存布局的策略：在 CUDA/ROCm 上默认启用，其他平台自动关闭，并保留环境变量 SGLANG_DIFFUSION_VAE_CHANNELS_LAST_3D 用于试验和回滚。同时新增了组件准确性测试套件，覆盖 1 GPU 和 2 GPU 场景，确保布局一致性。

功能与动机

PR #25985 修复了 Wan VAE 在 Conv3d weights 使用 channels_last_3d 时的正确性问题。此后续 PR 旨在将默认行为保持为 CUDA/ROCm 开启，同时为其他平台（如 NPU、CPU）提供安全 fallback，并保留显式环境变量用于试验和回滚。

实现拆解

1. 平台感知的支持判断：在 wan_common_utils.py 中添加 _channels_last_3d_supported_by_platform()，仅当 CUDA/ROCm 且 torch 支持 channels_last_3d 时返回 True。
2. 自动选择逻辑：在 vae_loader.py 中添加 _should_use_channels_last_3d()，根据环境变量和平台决定是否启用；同时修改 load_customized 方法中的两处条件，替换原来直接检查 torch.cuda.is_available() 和 envs.SGLANG_DIFFUSION_VAE_CHANNELS_LAST_3D 的代码。
3. 测试基础设施：在 component_accuracy.py 中添加 Conv3dLayoutStats 数据类和 _record_conv3d_layouts 上下文管理器，用于监控 Conv3d 调用的布局；添加 run_vae_channels_last_3d_parity 静态方法，分别以禁用和启用 channels_last_3d 加载 VAE，比较输出是否匹配。
4. 准确性测试集成：在 test_component_accuracy_1_gpu.py 和 test_component_accuracy_2_gpu.py 中添加对应的参数化测试类，指定特定 case（wan2_1_t2v_1.3b 和 wan2_2_i2v_a14b_2gpu）运行 parity 检查。
5. 单元测试：在 test_vae_loader.py 中添加两个测试，验证 match_conv3d_input_format 在非 CUDA 平台跳过转换，在 CUDA 上执行转换。

VAE 加载器中的自动选择逻辑（vae_loader.py）

```
import os
import torch.nn as nn
from sglang.multimodal_gen.runtime.utils.common import get_bool_env_var
```

```
from sglang.multimodal_gen.runtime.platforms import current_platform

VAE_CHANNELS_LAST_3D_ENV = "SGLANG_DIFFUSION_VAE_CHANNELS_LAST_3D"

def _convert_conv3d_weights_to_channels_last_3d(module: nn.Module) -> int:
    """将模块中所有 Conv3d 权重转换为 channels_last_3d 格式
    Returns: 转换的 Conv3d 模块数量
    """
    if not hasattr(torch, "channels_last_3d"):
        return 0
    num_converted = 0
    for m in module.modules():
        if isinstance(m, nn.Conv3d):
            try:
                m.weight.data = m.weight.data.to(memory_format=torch.channels_last_3d)
                num_converted += 1
            except Exception:
                continue
    return num_converted

def _should_use_channels_last_3d(server_args: ServerArgs, component_name: str) -> bool:
    """判断给定组件是否应使用 channels_last_3d
    仅在 CUDA/ROCm 平台且组件为 vae/video_vae 时返回 True;
    环境变量 SGLANG_DIFFUSION_VAE_CHANNELS_LAST_3D 可覆盖:
    - unset/auto: 启用 (True)
    - false: 禁用
    - true: 启用
    """
    if component_name not in ("vae", "video_vae") or not (
        current_platform.is_cuda() or current_platform.is_rocm()
    ):
        return False
    override = os.getenv(VAE_CHANNELS_LAST_3D_ENV)
    if override is None or override.strip().lower() == "auto":
        return True
    return get_bool_env_var(VAE_CHANNELS_LAST_3D_ENV)
```

准确性测试中的布局监控工具（component_accuracy.py）

```
from contextlib import contextmanager
from dataclasses import dataclass

VAE_CHANNELS_LAST_3D_ENV = "SGLANG_DIFFUSION_VAE_CHANNELS_LAST_3D"
VAE_CHANNELS_LAST_3D_PARITY_THRESHOLD = float(
    os.getenv("SGLANG_DIFFUSION_VAE_CHANNELS_LAST_3D_PARITY_THRESHOLD", "0.999")
)

@contextmanager
def _temporary_vae_channels_last_3d(enabled: bool):
    """临时设置环境变量以启用或禁用 channels_last_3d"""
```

```
    previous = os.environ.get(VAE_CHANNELS_LAST_3D_ENV)
    os.environ[VAE_CHANNELS_LAST_3D_ENV] = "true" if enabled else "false"
    try:
        yield
    finally:
        if previous is None:
            os.environ.pop(VAE_CHANNELS_LAST_3D_ENV, None)
        else:
            os.environ[VAE_CHANNELS_LAST_3D_ENV] = previous

@dataclass
class Conv3dLayoutStats:
    """记录 Conv3d 调用中的内存布局统计信息"""
    calls: int = 0
    channels_last_input_calls: int = 0
    channels_last_weight_calls: int = 0
    mixed_layout_calls: int = 0 # weight channels_last 但 input 不是

@contextmanager
def _record_conv3d_layouts():
    """监控 torch.nn.functional.conv3d 调用的布局"""
    stats = Conv3dLayoutStats()
    original_conv3d = torch.nn.functional.conv3d
    def wrapped_conv3d(input, weight, *args, **kwargs):
        if (
            isinstance(input, torch.Tensor)
            and isinstance(weight, torch.Tensor)
            and input.dim() == 5
            and weight.dim() == 5
            and hasattr(torch, "channels_last_3d")
        ):
            input_channels_last = input.is_contiguous(memory_format=torch.channels_last_3d)
            weight_channels_last = weight.is_contiguous(memory_format=torch.channels_last_3d)
        else:
            input_channels_last = False
            weight_channels_last = False
        stats.calls += 1
        stats.channels_last_input_calls += int(input_channels_last)
        stats.channels_last_weight_calls += int(weight_channels_last)
        stats.mixed_layout_calls += int(weight_channels_last and not input_channels_last)
        return original_conv3d(input, weight, *args, **kwargs)
    torch.nn.functional.conv3d = wrapped_conv3d
    try:
        yield stats
    finally:
        torch.nn.functional.conv3d = original_conv3d
```

评论区精华

本 PR 无有效 review 讨论，创作者自行合并。

风险与影响

- 平台兼容性：非 CUDA/ROCm 平台（如 NPU、CPU）不会启用 channels_last_3d，避免了之前出现的正确性问题。但环境变量强制覆盖可能导致意外，需用户自行保证平台支持。
- 浮点阈值：Parity 测试使用了 0.999 的阈值，可能因精度差异产生误报，但可从环境变量 SGLANG_DIFFUSION_VAE_CHANNELS_LAST_3D_PARITY_THRESHOLD 调整。
- 回归防护：新增的 1 GPU 和 2 GPU 准确性测试可在 CI 中持续运行，防止布局相关的数值漂移。

关联脉络

- PR #25985 修复了 Wan VAE 的 channels_last_3d 解码错误，本 PR 在此基础上进一步优化默认策略并完善测试覆盖率。
- 该系列 PR 体现了对扩散模型 VAE 组件内存布局的持续改进，从 bug 修复到主动防御，提升框架在不同硬件上的鲁棒性。