PR #45057 完整报告

vllm-project/vllm

[Bugfix] Handle HWC images in ImageProcessorItems.get_image_size

合并时间：2026-06-10 13:35

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/45057

执行摘要

- 一句话：修复 HWC 图片尺寸读取错误
- 推荐动作：推荐精读：这是一个简洁但典型的布局敏感性 bug 修复，展示了多模态推理中张量布局的隐式假设如何导致难以排查的运行时错误。建议关注 get_image_size 和 get_frame_size 保持一致的策略，以及参数化测试的设计模式。

功能与动机

ImageProcessorItems.get_image_size() 无条件将数组形状拆解为 CHW 格式：_, h, w = image.shape，当用户传入 HWC 图像（如 np.array(PIL.Image) 产生的形状 (480, 640, 3)）时，返回 ImageSize(width=3, height=640) 而非正确的 ImageSize(width=640, height=480)。该函数被多个多模态模型（InternVL、Gemma3、Molmo、H2OVL、Mistral3 等）用于计算图像特征 / 占位符 token 数，错误的尺寸会导致占位符数量与实际编码器输出不匹配，引发推断失败。同兄弟方法 VideoProcessorItems.get_frame_size() 的相同 bug 已在 #44509 中修复，但 get_image_size() 未被更新。

实现拆解

实现步骤：

1. 新增 HWC 检测逻辑：在 vllm/multimodal/parse.py 的 ImageProcessorItems.get_image_size 方法中，对 np.ndarray 或 torch.Tensor 类型，通过 ndim == 3 and shape[-1] in (1, 3, 4) 判断是否为 HWC 布局，若是则从 shape[0] 和 shape[1] 获取高和宽，否则使用原 CHW 布局的拆解方式。该检测逻辑与 #44509 中修复 get_frame_size 时使用的完全一致。
2. 新增测试文件：创建 tests/multimodal/test_parse.py，使用参数化测试覆盖 PIL 图像、HWC numpy/torch 数组、CHW numpy/torch 数组共 5 种输入，验证 get_image_size 和 get_frame_size 均返回正确的 (width, height) 元组。
3. AI 贡献归属：在最后一次提交中添加了 AI 共同作者署名（Claude Code 用于定位 bug 和起草代码）。

关键文件：

- vllm/multimodal/parse.py（模块 解析器；类别 source；类型 core-logic；符号 get_image_size）：核心修复文件。在 ImageProcessorItems.get_image_size 中增加 HWC 布局检测，是 bug 的直接修复点。

- tests/multimodal/test_parse.py（模块 解析器；类别 test；类型 test-coverage；符号 test_image_size_hwc_chw, test_frame_size_hwc_chw）：新增测试文件，参数化覆盖 PIL/HWC numpy/HWC torch/CHW numpy/CHW torch 五种输入，验证 get_image_size 和 get_frame_size 的正确性，是修复的质量保证。

关键符号：get_image_size, test_image_size_hwc_chw, test_frame_size_hwc_chw

关键源码片段

vllm/multimodal/parse.py

核心修复文件。在 ImageProcessorItems.get_image_size 中增加 HWC 布局检测，是 bug 的直接修复点。

```
# vllm/multimodal/parse.py

class ImageProcessorItems(ProcessorBatchItems[HfImageItem | None]):
    # ... 其他方法省略 ...

    def get_image_size(self, item_idx: int) -> ImageSize:
        image = self.get(item_idx)
        if image is None:
            raise ValueError(f"Cannot get size of cached image at {item_idx}")

        if isinstance(image, PILImage.Image):
            return ImageSize(*image.size)
        if isinstance(image, (np.ndarray, torch.Tensor)):
            # 检测 HWC 格式（例如来自 np.array(PIL.Image)）
            # 特征：最后一维通道数为 1、3 或 4
            if image.ndim == 3 and image.shape[-1] in (1, 3, 4):
                # HWC 布局：高、宽分别位于下标 0 和 1
                h, w = image.shape[0], image.shape[1]
            else:
                # CHW 布局（标准 PyTorch / numpy 约定），通道下标 0
                _, h, w = image.shape
            return ImageSize(w, h)

        assert_never(image)
```

tests/multimodal/test_parse.py

新增测试文件，参数化覆盖 PIL/HWC numpy/HWC torch/CHW numpy/CHW torch 五种输入，验证 get_image_size 和 get_frame_size 的正确性，是修复的质量保证。

```
# tests/multimodal/test_parse.py

import numpy as np
import pytest
import torch
from PIL import Image
```

```
from vllm.multimodal.parse import ImageProcessorItems, VideoProcessorItems

H, W = 480, 640


@pytest.mark.parametrize(
    "image",
    [
        Image.new("RGB", (W, H)), # PIL 图像，始终是 HWC
        np.zeros((H, W, 3), dtype=np.uint8), # HWC numpy 数组
        torch.zeros((H, W, 3), dtype=torch.uint8), # HWC torch 张量
        np.zeros((3, H, W), dtype=np.uint8), # CHW numpy 数组
        torch.zeros((3, H, W), dtype=torch.uint8), # CHW torch 张量
    ],
)
def test_image_size_hwc_chw(image):
    """验证 `get_image_size` 对 HWC 和 CHW 布局均返回正确宽度和高度。"""
    items = ImageProcessorItems([image])
    assert items.get_image_size(0) == (W, H) # (width, height)


@pytest.mark.parametrize(
    "frame",
    [
        Image.new("RGB", (W, H)),
        np.zeros((H, W, 3), dtype=np.uint8),
        torch.zeros((H, W, 3), dtype=torch.uint8),
        np.zeros((3, H, W), dtype=np.uint8),
        torch.zeros((3, H, W), dtype=torch.uint8),
    ],
)
def test_frame_size_hwc_chw(frame):
    """验证 `get_frame_size` 与 `get_image_size` 行为一致。"""
    items = VideoProcessorItems([[frame]])
    assert items.get_frame_size(0) == (W, H)
```

评论区精华

核心讨论是要求添加 AI 贡献归属。项目维护者 DarkLight1337 要求提交者添加 AI attribution via co-authors，提交者 YellowFoxH4XOR 随即在第三次提交中增加了 Co-authored-by: Claude 署名。评审者之后批准了 PR，未涉及技术争议。

- AI 贡献归属 (other): 提交者添加了 Co-authored-by: Claude 署名，PR 随后被批准。

风险与影响

- 风险：风险极低。变更仅修改一个方法中的 6 行代码，逻辑与已合并的 #44509 中视频路径的修复完全一致，是一个成熟且经过验证的模式。新增的测试覆盖了 HWC 和 CHW 两种布局以及 PIL 输入，防止回归。

- 影响：影响范围集中在对 ImageProcessorItems.get_image_size 的调用者，即所有使用该函数计算图像占位符数量的多模态模型（InternVL、Gemma3、Molmo、H2OVL、Mistral3 等）。修复后，使用 HWC 格式输入（如直接从 PIL Image 转换的 numpy 数组）时，占位符数量正确，避免了 EngineDeadError 或静默的嵌入数量不匹配问题。对已使用 CHW 格式或 PIL Image 的用户无影响。
- 风险标记：低风险

关联脉络

- PR #44509 [Bugfix] MiniCPM-V-4.6 video inference crash: placeholder count mismatches visual embedding count: 修复了相同的 HWC 布局 bug 在视频路径（get_frame_size）中的表现，当前 PR 是图像路径的对等修复，使用了完全相同的检测模式。