PR #44609 完整报告

vllm-project/vllm

Support MiniCPMV batched preprocessing

合并时间：2026-06-05 23:05

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/44609

执行摘要

- 一句话：MiniCPMV 批量预处理支持
- 推荐动作：建议精读，尤其是 pad 方法的实现和 _convert_images_texts_to_inputs 中批处理循环的设计。对于维护 vendored processor 的团队，此 PR 展示了如何在保持兼容性的同时扩展功能，值得借鉴。

功能与动机

修复 MiniCPMV 模型批量推理时的崩溃：text=['(<image>./</image>)'] + images=[[img]] 格式导致处理器报错。动机来自 Issue 和 PR body 指出的实际使用错误，以及上游 MiniCPM-V 处理器已支持批处理但 vLLM 的 vendored 版本未跟进。

实现拆解

1. 调整 _convert 返回值：移除 input_ids.unsqueeze(0)，使输出保持 1D 张量，以便后续批处理时自由组合。
2. 重写 _convert_images_texts_to_inputs 支持批处理：将原来仅处理单个文本字符串的逻辑改为循环处理 texts 列表。对每个文本，解析图像标签、生成占位符文本、调用 _convert 获取 input_ids 和 image_bounds，并分别收集到列表中。
3. 新增通用 pad 方法：替代原来的 pad（只支持单键），新的方法接受 inputs 列表，对序列进行左填充，并返回填充后的张量及每个样本的填充长度。该方法参考了上游 openbmb/MiniCPM-V-4_5 的 processing_minicpmv.py 实现。
4. 整合输出：在 return MiniCPMVBatchFeature 中，使用 padded_input_ids，添加 attention_mask（基于非零位置），并将 image_bound 改为列表形式（每个元素是偏移后的 image_bounds）。同时更新了 __call__ 的 docstring 声明支持批量输入。

关键文件：

- vllm/transformers_utils/processors/minicpmv.py（模块 处理器；类别 source；类型 core-logic；符号 pad, _convert, _convert_images_texts_to_inputs, call）：唯一的修改文件，包含所有核心逻辑变更：批处理支持、pad 方法重写、输出结构调整。

关键符号：pad, _convert, _convert_images_texts_to_inputs

关键源码片段

vllm/transformers_utils/processors/minicpmv.py

唯一的修改文件，包含所有核心逻辑变更：批处理支持、pad 方法重写、输出结构调整。

```
# vllm/transformers_utils/processors/minicpmv.py
# 关键改动：_convert 移除批次维度，支持批量处理

def _convert(self, input_str, max_inp_length: int | None = None):
    # ... 原有逻辑 ...
    input_ids = torch.tensor(input_ids, dtype=torch.int32)
    # ... 计算 image_bounds ...
    # 之前返回 input_ids.unsqueeze(0), image_bounds
    return input_ids, image_bounds # 保持 1D，由调用方组合批量

def _convert_images_texts_to_inputs(self, images, texts, do_pad=False, ...):
    # ...
    if isinstance(texts, str):
        texts = [texts] # 统一为列表

    input_ids_list = []
    image_bounds_list = []
    for index, text in enumerate(texts):
        # 对每个文本单独处理（解析占位符、生成 input_ids、获取 image_bounds）
        # ...（原有单样本逻辑移至循环内）...
        input_ids, image_bounds = self._convert(final_text, max_length)
        input_ids_list.append(input_ids)
        image_bounds_list.append(image_bounds)

    # 左填充 batch
    padded_input_ids, padding_lengths = self.pad(
        input_ids_list, padding_side="left",
    )
    # 根据填充长度偏移 image_bounds
    for i, length in enumerate(padding_lengths):
        image_bounds_list[i] = image_bounds_list[i] + length

    return MiniCPMVBatchFeature(data={
        "input_ids": padded_input_ids,
        "attention_mask": padded_input_ids.ne(0), # 新增 attention_mask
        "pixel_values": images_val,
        "image_sizes": image_sizes,
        "image_bound": image_bounds_list, # 改为列表
        "tgt_sizes": tgt_sizes,
    })

# 新增通用 pad 方法（参考 upstream）
def pad(self, inputs, max_length=None, padding_value=0, padding_side="left"):
    """
    对多个序列进行填充，返回填充后的张量和每个序列的填充长度。
    主要用于批量预处理中的左填充。
```

```
    """
    # ... 实现细节：确定 max_length，左侧填充 padding_value，记录长度差
```

评论区精华

reviewer tc-mb 本地验证了修复的正确性，确认批量推理和视频推理均正常。另一个讨论点涉及移除 max_transformers_version="4.57" 导致的 CI 失败，作者建议暂时回退该改动以先合入功能修复，maintainer DarkLight1337 同意。最终合并版本仅包含处理器逻辑改动。

- 本地验证批处理和视频推理 (testing): 确认变更正确，无回归。
- 移除 transformers 版本限制导致 CI 失败 (other): 撤销版本限制改动，仅合并处理器逻辑。

风险与影响

- 风险：
  1. 回归风险：_convert 移除了 unsqueeze(0)，可能影响依赖该维度假设的其他调用方。但 _convert 是私有方法，且所有调用路径均在本次更新中一并调整，风险可控。
  2. 测试覆盖不足：本次未添加新测试文件，仅依赖现有 CI 和手动验证。若后续有边缘批量场景（如空文本、不同数量图像）可能漏测。
  3. 上游同步风险：pad 方法参考了上游实现，但未做完全同步，未来上游更新可能导致行为差异。 - 影响：用户影响：使用 MiniCPMV 模型的用户现在可以正常进行批量图像推理，之前被阻塞的工作流得到修复。系统影响：仅影响 MiniCPMV 模型路径，不影响其他模型或全局行为。团队影响：为其他类似模型的批量预处理支持提供了可参考的模板，尤其是 vendored processor 的改造方式。
- 风险标记：缺少测试覆盖 , 上游同步依赖

关联脉络

- 暂无明显关联 PR