PR #44282 完整报告

vllm-project/vllm

[Bugfix] Vendor MiniCPMV/MiniCPMO processors to unblock Transformers v5

合并时间：2026-06-02 22:14

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/44282

执行摘要

- 一句话：Vendor MiniCPMV/MiniCPMO 处理器以解锁 Transformers v5 升级
- 推荐动作：建议开发者关注 vendor 处理器与上游的差异，确保后续 Transformers 升级时及时同步更新。此 PR 采用的 vendor 策略（直接复制关键依赖）适用于其他类似场景，但需评估长期维护成本。同时，建议增加更多端到端测试以覆盖新处理器的各种输入组合。

功能与动机

Transformers v5 升级后，MiniCPMV 和 MiniCPMO 的原始处理器因使用了已删除的 API（如 tokenizer.im_start_id）而失败，触发 AttributeError。此 PR 通过将处理器代码内部化，解除了升级阻塞，解决了 Issue #38385 和 #30566 中的相关 skip 标记。

实现拆解

实现分为以下步骤：

1. 新增 vendor 处理器文件：在 vllm/transformers_utils/processors/ 下新建 minicpmv.py（+314 行）和 minicpmo.py（+603 行），分别实现 MiniCPMVProcessor 和 MiniCPMOProcessor 类。它们继承自 transformers.ProcessorMixin，并提供 __call__、batch_decode、decode 等方法，与上游 API 保持一致。特别注意对音频、图像等多模态输入的处理逻辑。
2. 修改模型代码以使用 vendor 处理器：在 vllm/model_executor/models/minicpmv.py 和 minicpmo.py 的 get_hf_processor 方法中，导入并使用 vendor 处理器替换原有的 HF 处理器。同时保留对 numpy 数组序列化的兼容处理（将 mean/std 等属性从 numpy 数组转换为列表）。
3. 注册 vendor 处理器：在 vllm/transformers_utils/processors/__init__.py 中将新处理器类添加到 _import_structure 字典中，确保可通过名称查找。
4. 移除测试跳过标记：删除 tests/lora/test_minicpmv_tp.py 中因 Transformers v5 而设置的 pytest.mark.skipif 条件，以及 tests/models/multimodal/generation/test_common.py 中对 minicpmv_25 和 minicpmo_26 的跳过标记（从注释改为完整删除）。现在相关测试可在 Transformers v5 上正常执行。
5. 处理 Review 反馈：修复了 MiniCPMVProcessor 中当 images 为 None 时可能出现的 UnboundLocalError，调整了 docstring 以满足 mkdocs strict 模式，并修复了类型检查问题。

关键文件：

- vllm/transformers_utils/processors/minicpmo.py（模块 处理器；类别 source；类型 dependency-wiring；符号 MiniCPMOProcessor, MiniCPMOProcessor.init, MiniCPMOProcessor.call, MiniCPMOProcessor.get_audio_placeholder）：MiniCPMO 处理器核心实现，包含音频特征提取和多模态输入处理
- vllm/transformers_utils/processors/minicpmv.py（模块 处理器；类别 source；类型 dependency-wiring；符号 MiniCPMVProcessor, MiniCPMVProcessor.init, MiniCPMVProcessor.call, MiniCPMVProcessor.batch_decode）：MiniCPMV 处理器核心实现，包含图像处理和 tokenization 功能
- vllm/model_executor/models/minicpmo.py（模块 模型层；类别 source；类型 data-contract；符号 get_hf_processor）：修改 get_hf_processor 方法以使用 vendor 处理器，并处理 numpy 数组序列化
- vllm/model_executor/models/minicpmv.py（模块 模型层；类别 source；类型 data-contract；符号 get_hf_processor）：修改 get_hf_processor 方法以使用 vendor 处理器
- vllm/transformers_utils/processors/__init__.py（模块 配置；类别 source；类型 core-logic）：注册 vendor 处理器到查找表
- tests/lora/test_minicpmv_tp.py（模块 测试；类别 test；类型 test-coverage）：移除了因 Transformers v5 不兼容导致的测试跳过标记
- tests/models/multimodal/generation/test_common.py（模块 测试；类别 test；类型 test-coverage）：移除了 minicpmv_25 和 minicpmo_26 的 skip 标记

关键符号：MiniCPMVProcessor.init, MiniCPMVProcessor.call, MiniCPMVProcessor.batch_decode, MiniCPMVProcessor.decode, MiniCPMOProcessor.init, MiniCPMOProcessor.call, MiniCPMOProcessor.get_audio_placeholder, MiniCPMOProcessor.audio_feature_extract, MiniCPMOProcessingInfo.get_hf_processor, MiniCPMVProcessingInfo.get_hf_processor

关键源码片段

vllm/transformers_utils/processors/minicpmo.py

MiniCPMO 处理器核心实现，包含音频特征提取和多模态输入处理

```
class MiniCPMOProcessor(ProcessorMixin):
    """MiniCPMO 处理器，包装图像处理器、特征提取器和分词器。"""
    def __init__(self, image_processor=None, feature_extractor=None, tokenizer=None, pool_step=
    2):
        # 调用父类初始化，设置 component
        super().__init__(image_processor, feature_extractor, tokenizer)
        self.version = image_processor.version
        self.pool_step = pool_step # 音频 pooling 步长

    def __call__(self, text, images=None, audios=None, audio_parts=None,
                 max_length=None, do_pad=True, max_slice_nums=None,
                 use_image_id=True, chunk_input=False, return_tensors=TensorType.PYTORCH,
```

```
            sampling_rate=16000, **kwargs):
        # 处理图像：若 images 不为 None，调用 image_processor 生成图像输入
        if images is not None:
            image_inputs = self.image_processor(images, do_pad=do_pad,
                                                max_slice_nums=max_slice_nums,
                                                return_tensors=return_tensors)
        else:
            image_inputs = None

        # 处理音频：调用音频特征提取方法
        if audios is not None:
            audio_features, audio_feature_lens, audio_phs = self.audio_feature_extract(
                audios, audio_parts, chunk_input, sampling_rate)
        else:
            audio_features, audio_feature_lens, audio_phs = [], [], []

        # 合并图像、音频占位符和文本到模型输入
        model_inputs = self._convert_omni_to_inputs(
            image_inputs, audio_phs, text,
            max_slice_nums=max_slice_nums, use_image_id=use_image_id,
            max_length=max_length, **kwargs)

        model_inputs["audio_features"] = audio_features
        model_inputs["audio_feature_lens"] = audio_feature_lens

        return MiniCPMOBatchFeature(data={**model_inputs})
```

评论区精华

在 Review 中，@DarkLight1337 要求彻底删除 tests/models/multimodal/generation/test_common.py 中注释掉的 marks=[pytest.mark.skip(...)] 行，而非仅注释。作者 @wjinxu 已按请求完整删除，并最终获得批准。其余讨论涉及测试通过和 CI 问题，作者指出一个无关的 CI 失败（tracing 测试 segfault），@DarkLight1337 强制合并。

- 删除测试中的跳过标记 (testing): 已删除并批准合并。

风险与影响

- 风险：
  - vendor 代码同步成本：将上游代码复制到内部后，需要与 Hugging Face 的更新保持同步，否则可能错过重要修复或功能改进。
  - 潜在的兼容性问题：新处理器虽然复制自上游，但可能未覆盖所有边界情况，特别在音频处理和图像分片方面。
  - 测试覆盖：虽然移除了跳过标记，但当前测试用例可能不足以覆盖所有功能路径。建议后续增加更多针对性测试。
- 影响：

- 用户影响：使用 MiniCPMV 和 MiniCPMO 模型的用户可以正常升级到 Transformers v5，不再受阻塞。
- 系统影响：vLLM 不再依赖 Hugging Face Transformers 中这两个处理器的特定版本，提高了向后兼容性。但需要额外维护约 917 行 vendor 代码。
- 团队影响：解除了 Transformers v5 升级的一个关键障碍，推动项目整体升级。
- 风险标记：vendor 代码同步，潜在兼容性问题，测试覆盖不足

关联脉络

- PR #38437 Vendor MiniCPMV/MiniCPMO processors: 此 PR 是从 38437 的原始提交 rebase 而来，保留了作者签名
- PR #30566 Update to transformers v5: 此 PR 解除了升级到 Transformers v5 的一个阻塞
- PR #38385 MiniCPMV cannot apply processor: 此 PR 修复了该 Issue 描述的 MiniCPMV 处理器失败问题