PR #39869 完整报告

vllm-project/vllm

[Model] Use mm_features for Keye-VL and Keye-1.5-VL M-RoPE

合并时间：2026-04-16 17:16

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/39869

执行摘要

- 一句话：为 Keye-VL 和 Keye-1.5-VL 模型重构 M-RoPE 位置计算，切换到 mm_features 驱动。
- 推荐动作：此 PR 值得精读，特别是 iter_mm_grid_thw 的设计展示了如何从传统 token 处理过渡到基于元数据的多模态接口。关注视频拆分逻辑和测试用例的构造，以理解 M-RoPE 计算的关键细节。

功能与动机

根据 PR body，目的是实现 Keye-VL 和 Keye-1.5-VL 的 M-RoPE 计算重构，以便从 mm_feature.mm_position 和网格元数据派生多模态跨度，而不是从 token ID 重建图像和视频区域。这有助于统一接口，简化逻辑，并为后续多模态特性开发铺平道路。

实现拆解

1. 入口方法重构：在 keye.py 和 keye_vl1_5.py 中，get_mrope_input_positions 方法不再调用 MultiModalFeatureSpec.gather_kwargs，而是通过新增的 iter_mm_grid_thw 遍历排序后的 mm_features，获取每个多模态块的偏移和网格尺寸。
2. 核心迭代器实现：新增 iter_mm_grid_thw 方法，接受 mm_features 列表，根据 mm_position.offset 排序，为图像和视频模态生成 (offset, t, h, w) 元组。图像直接提取网格尺寸；视频使用 _split_video_grid_thw 拆分为帧级行。
3. 视频网格拆分：在 keye.py 中新增静态方法 _split_video_grid_thw，将视频网格沿时间维度拆分为 [1, h, w] 的行列表，以保持 Keye 当前的帧级 M-RoPE 行为。
4. 测试配套：新增 tests/model_executor/test_keye_mrope.py 和 test_keye_vl1_5_mrope.py，使用虚拟配置和特征模拟文本、单图像、图像视频交错等场景，通过断言验证位置计算和 delta 值的正确性。
5. 导入和结构调整：在两个模型文件中添加 Iterator 导入，并调整控制流以处理新的数据契约，确保代码可读性和维护性。

关键文件：

- vllm/model_executor/models/keye.py（模块 模型层；类别 source；类型 core-logic；符号 _split_video_grid_thw, iter_mm_grid_thw, get_mrope_input_positions）：Keye-VL 模型的核心文件，重构了 M-RoPE 位置计算方法，新增视频拆分和多模态迭代器。

- vllm/model_executor/models/keye_vl1_5.py（模块 模型层；类别 source；类型 core-logic；符号 iter_mm_grid_thw, get_mrope_input_positions）：Keye-1.5-VL 模型的核心文件，类似重构 M-RoPE 计算，特别处理了嵌入范围以保持视频行为。
- tests/model_executor/test_keye_mrope.py（模块 测试模块；类别 test；类型 test-coverage；符号 _force_cpu_default_device, DummyVisionConfig, DummyConfig, make_model）：新增 Keye-VL 的 CPU 单元测试，验证重构后 M-RoPE 位置计算的正确性。
- tests/model_executor/test_keye_vl1_5_mrope.py（模块 测试模块；类别 test；类型 test-coverage；符号 _force_cpu_default_device, DummyVisionConfig, DummyConfig, make_model）：新增 Keye-1.5-VL 的 CPU 单元测试，特别验证视频嵌入范围的使用。

关键符号：iter_mm_grid_thw, _split_video_grid_thw, get_mrope_input_positions

关键源码片段

vllm/model_executor/models/keye.py

Keye-VL 模型的核心文件，重构了 M-RoPE 位置计算方法，新增视频拆分和多模态迭代器。

```
def iter_mm_grid_thw(
    self, mm_features: list[MultiModalFeatureSpec]
) -> Iterator[tuple[int, int, int, int]]:
    spatial_merge_size = self.config.vision_config.spatial_merge_size #
    获取空间合并尺寸以调整网格

    for mm_feature in sorted(mm_features, key=lambda f: f.mm_position.offset): #
    按偏移排序确保顺序
        if mm_feature.data is None:
            raise ValueError("M-RoPE calculation requires multimodal feature data") #
            数据缺失则报错

        if mm_feature.modality == "image":
            grid_thw = mm_feature.data["image_grid_thw"].data # 提取图像网格数据
            if isinstance(grid_thw, torch.Tensor):
                if grid_thw.ndim == 2:
                    assert grid_thw.shape[0] == 1 # 确保为单图像
                    t, h, w = grid_thw[0].tolist() # 转换为列表格式
                else:
                    t, h, w = grid_thw.tolist()
            else:
                # 处理非 Tensor 情况，当前为死代码，可能为历史遗留
                if isinstance(grid_thw[0], list):
                    assert len(grid_thw) == 1
                    t, h, w = grid_thw[0]
                else:
                    t, h, w = grid_thw

            yield (
                mm_feature.mm_position.offset, # 多模态块在输入序列中的起始偏移
                t,
```

```
                h // spatial_merge_size, # 调整高度以适应 LLM 网格
                w // spatial_merge_size, # 调整宽度以适应 LLM 网格
            )
        elif mm_feature.modality == "video":
            current_offset = mm_feature.mm_position.offset
            for t, h, w in self._split_video_grid_thw( # 拆分视频网格为帧级
                mm_feature.data["video_grid_thw"].data
            ):
                llm_grid_h = h // spatial_merge_size
                llm_grid_w = w // spatial_merge_size
                yield (current_offset, t, llm_grid_h, llm_grid_w) # 为每帧生成元组
                current_offset += t * llm_grid_h * llm_grid_w # 更新偏移以处理连续帧
        else:
            raise ValueError(f"Unsupported modality: {mm_feature.modality}") # 不支持其他模态
```

评论区精华

- HACK 注释问题：gemini-code-assist[bot] 指出测试文件中的 # HACK. 注释可能隐藏维护问题，但此问题未在 PR 中解决。
- 死代码移除：同一评论者建议移除 keye.py 中处理非 Tensor 图像网格的 else 块，认为它是死代码，但 PR 中未采纳。
- Keye-1.5-VL 扩展：DarkLight1337 最初要求将此重构扩展到 Keye-1.5-VL，作者随后添加并验证，使用 lm-eval 显示性能无显著变化，最终批准 PR。
  - 测试文件中的 HACK 注释 (style): 未在 PR 中解决，遗留为潜在技术债。
  - keye.py 中的死代码 (correctness): PR 中未采纳，代码保留但可能不影响功能。
  - 扩展 Keye-1.5-VL 支持 (design): 成功扩展，验证通过后 PR 获批准。

风险与影响

- 风险：- 回归风险：M-RoPE 计算逻辑变更可能导致位置编码错误，影响模型输出质量，尤其是在视频处理中，需要确保 iter_mm_grid_thw 生成的偏移和网格尺寸准确。
- 兼容性风险：新代码强依赖 mm_features 的特定结构（如 mm_position.offset 和 data["image_grid_thw"]），如果上游接口变化，可能导致运行时错误。
- 性能风险：新增的迭代和拆分操作可能引入轻微计算开销，但鉴于是重构且测试覆盖 CPU，实际推理影响应可控。
- 代码质量风险：review 中提到的 HACK 注释和死代码未处理，可能增加长期维护难度和潜在 bug。
- 影响：- 用户影响：使用 Keye-VL 或 Keye-1.5-VL 模型的开发者无需主动更改代码，但内部 M-RoPE 计算方式更新，应通过测试确保行为一致。
- 系统影响：代码更模块化，减少了基于 token 搜索的复杂性，使多模态特征处理更统一，有助于未来扩展。
- 团队影响：为基于 mm_features 的多模态模型开发提供了参考模式，促进代码标准化和团队协作效率。

- 风险标记：核心路径变更，数据契约依赖，未处理代码问题

关联脉络

- 暂无明显关联 PR