PR #40282 完整报告

vllm-project/vllm

Add Granite 4.1 Vision as built-in multimodal model

合并时间：2026-04-21 20:43

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/40282

执行摘要

- 一句话：新增 Granite 4.1 Vision 内置多模态模型支持，集成 SigLip 视觉编码器和深度堆叠特征注入。
- 推荐动作：该 PR 值得精读，特别是关注深层特征注入的设计（参考 Qwen3-VL 模式）和 Pipeline Parallelism 集成方式。建议工程师学习缓冲区管理和线程安全实践，以及如何处理尚未 upstream 的模型配置。

功能与动机

根据 PR body，添加内置支持是因为 'granite4_vision model type is not yet in the transformers version pinned by vLLM'，需要绕过此限制以加载模型。模型使用 SigLip2 视觉编码器和深度堆叠特征注入，以支持视觉语言任务。

实现拆解

1. 模型实现：在 vllm/model_executor/models/granite4_vision.py 中定义 Granite4VisionForConditionalGeneration 类，集成 Granite 语言骨干、SigLIP 视觉编码器，并实现深度堆叠特征注入和窗口 Q-Former 下采样器（如 InterpolateDownsampler、SpatialOffsetDownsampler、WindowQFormerDownsampler）。
2. 配置与处理器：新增 vllm/transformers_utils/configs/granite4_vision.py 和 vllm/transformers_utils/processors/granite4_vision.py，定义 Granite4VisionConfig 和 Granite4VisionProcessor 类，以支持模型加载和特征计算，处理 Hub 别名字段并覆盖特征计数方法。
3. 注册与集成：修改 vllm/model_executor/models/registry.py、vllm/transformers_utils/config.py 等文件，将模型注册到 _CONFIG_REGISTRY 和 _MULTIMODAL_MODELS，确保 vLLM 能识别和加载模型。
4. 测试配套：更新 tests/models/multimodal/generation/test_common.py 添加测试用例，包括 HF vs vLLM 输出比较和 LoRA 适配器支持；同时更新 tests/models/registry.py 和 examples/offline_inference/vision_language_multi_image.py 以提供示例和覆盖率。
5. 文档更新：修改 docs/models/supported_models.md 添加模型到支持列表，保持文档同步。

关键文件：

- vllm/model_executor/models/granite4_vision.py（模块 模型执行器；类别 source；类型 core-logic；符号 InterpolateDownsampler, SpatialOffsetDownsampler,

WindowQFormerDownsampler, Granite4VisionForConditionalGeneration）：模型实现核心文件，定义了 Granite4VisionForConditionalGeneration 类和深层特征注入逻辑。

- vllm/transformers_utils/processors/granite4_vision.py（模块 Transformers 工具；类别 source；类型 core-logic；符号 Granite4VisionProcessor, _get_number_of_features）：处理器类实现，扩展 LlavaNextProcessor 以处理窗口 Q-Former 下采样时的特征计数。
- vllm/transformers_utils/configs/granite4_vision.py（模块 Transformers 工具；类别 source；类型 core-logic；符号 Granite4VisionConfig）：配置类定义，允许 vLLM 加载尚未在 upstream transformers 中支持的模型类型。
- examples/offline_inference/vision_language_multi_image.py（模块 离线推理；类别 source；类型 entrypoint；符号 load_granite4_vision）：示例文件更新，添加 Granite 4.1 Vision 模型的离线推理示例。
- tests/models/multimodal/generation/test_common.py（模块 通用测试；类别 test；类型 test-coverage；符号 _granite4_vision_vllm_to_hf_output）：测试文件更新，添加 Granite 4.1 Vision 的 HF vs vLLM 输出比较测试。

关键符号：InterpolateDownsampler.call, SpatialOffsetDownsampler.call, WindowQFormerDownsampler.forward, Granite4VisionProcessor._get_number_of_features, Granite4VisionConfig.init, Granite4VisionForConditionalGeneration.forward

关键源码片段

vllm/model_executor/models/granite4_vision.py

模型实现核心文件，定义了Granite4VisionForConditionalGeneration类和深层特征注入逻辑。

```
class Granite4VisionForConditionalGeneration(nn.Module):
    """vLLM implementation of Granite 4 Vision with deepstack feature injection."""

    def forward(self, hidden_states: torch.Tensor, deepstack_input_embeds:
    Optional[IntermediateTensors] = None) -> torch.Tensor:
        # 深度堆叠特征注入：将视觉特征添加到特定语言层
        if deepstack_input_embeds is not None:
            for layer_idx in self._ds_layer_indices:
                if layer_idx in deepstack_input_embeds:
                    features = deepstack_input_embeds[layer_idx]
                    # 使用就地加法避免克隆，提升性能并减少内存分配
                    hidden_states[self._vision_mask] += features[self._vision_mask]
        # 继续语言模型前向传播
        return self.language_model(hidden_states)
```

vllm/transformers_utils/processors/granite4_vision.py

处理器类实现，扩展 LlavaNextProcessor 以处理窗口 Q-Former 下采样时的特征计数。

```
class Granite4VisionProcessor(LlavaNextProcessor):
    """Processor for Granite 4 Vision with Window Q-Former downsampling support."""

    def _get_number_of_features(self, orig_height: int, orig_width: int, height: int, width: int) ->
    int:
```

```
        # 计算基础特征数，考虑下采样率
        patches_height = height // self.patch_size
        patches_width = width // self.patch_size
        if self.downsample_rate is not None:
            ds_rate = Fraction(self.downsample_rate)
            patches_height = int(patches_height * ds_rate) # 应用下采样率
            patches_width = int(patches_width * ds_rate)
        # 调用父类方法计算未填充和新行特征
        unpadded_features, newline_features = self._get_unpadded_features(...)
        base_features = patches_height * patches_width + self.num_additional_image_tokens
        return unpadded_features + newline_features + base_features
```

vllm/transformers_utils/configs/granite4_vision.py

配置类定义，允许 vLLM 加载尚未在 upstream transformers 中支持的模型类型。

```
class Granite4VisionConfig(transformers.PretrainedConfig):
    """Configuration for Granite 4 Vision model."""

    def __init__(self, vision_config: dict[str, Any] | None = None, text_config: dict[str, Any] |
    None = None, **kwargs):
        # 处理 Hub 别名字段以兼容不同命名约定
        self.deepstack_layer_map = kwargs.get('deepstack_layer_map') or kwargs.get('vision_
        layer_to_llm_layer')
        self.use_spatial_sampling = kwargs.get('use_spatial_sampling', False)
        # 初始化视觉和文本配置
        if vision_config is None:
            vision_config = {}
        self.vision_config = transformers.CONFIG_MAPPING.get(vision_config.get('model_type',
        'siglip_vision_model'), transformers.PretrainedConfig)(**vision_config)
        if text_config is None:
            text_config = {}
        self.text_config = transformers.CONFIG_MAPPING.get(text_config.get('model_type',
        'granite'), transformers.PretrainedConfig)(**text_config)
        super().__init__(**kwargs)
```

评论区精华

- 线程安全：gemini-code-assist[bot] 指出存储请求特定状态在模型属性中不安全，已通过引入 IntermediateTensors 和预分配 GPU 缓冲区解决，匹配 Qwen3-VL 模式。
- 性能优化：评论建议避免克隆操作，改为就地加法以提升效率，提交中已修复为使用 += 操作。
- Pipeline Parallelism 支持：发现 PP 实现中深层特征未传递到后续 ranks，通过修改 make_empty_intermediate_tensors 和缓冲区管理解决。
- LoRA 合并逻辑：手动 LoRA 合并被标记为脆弱且不兼容量化，最终移除了该逻辑，推荐使用 vLLM 原生 LoRA 服务（--enable-lora --default-mm-loras）。
- 测试逻辑错误：在 post-processor 中处理 idx == 0 时的错误，已添加显式检查修复。

- 代码风格与集成：DarkLight1337 建议按字母顺序排序测试条目、使用 vLLM 内置 Blip2QFormerModel，均已采纳。
  - 线程安全与状态管理 (correctness): 通过引入 IntermediateTensors 和预分配 GPU 缓冲区解决，匹配 Qwen3-VL 模式。
  - 性能优化与内存效率 (performance): 提交中修复为使用 += 操作，提升了效率。
  - Pipeline Parallelism 支持 (design): 通过修改 make_empty_intermediate_tensors 和缓冲区管理解决，确保特征跨 ranks 传递。
  - LoRA 合并逻辑 (design): 移除了该逻辑，推荐使用 vLLM 原生 LoRA 服务（--enable-lora --default-mm-loras）。
  - 测试逻辑错误 (correctness): 添加显式 idx == 0 检查修复逻辑。

风险与影响

- 风险：技术风险包括： 1) 线程安全风险：尽管已修复，但深层缓冲区管理仍需谨慎，避免在并发请求中状态污染； 2) 性能风险：特征注入循环可能在高批次大小下成为瓶颈，需监控内存使用； 3) 兼容性风险：模型配置处理 Hub 别名字段，若 upstream transformers 更新可能导致冲突； 4) 测试覆盖：模型尚未公开，测试依赖于 is_available_online=False，可能掩盖线上问题。
- 影响：影响范围： 1) 用户：可直接使用 Granite 4.1 Vision 模型进行多模态推理，无需等待 upstream 支持； 2) 系统：扩展 vLLM 多模态模型库，增强视觉语言任务能力； 3) 团队：新增代码需维护，但遵循现有模式（如 Qwen3-VL），降低长期负担。影响程度中等，主要限于模型集成层。
  - 风险标记：线程不安全初始化，性能瓶颈，PP 兼容性问题，量化不兼容，测试覆盖不足

关联脉络

- PR #39502 feat(multimodal): support externally processed mm_kwargs with cache injection: 同为多模态功能扩展，涉及缓存注入和特征处理，技术领域相关。
- PR #40411 [Bugfix] Gemma4: fix multimodal embedder norm order to match HF reference: 涉及多模态嵌入器修复，展示模型集成中的对齐问题和配置处理模式。