PR #40796 完整报告

vllm-project/vllm

[Bugfix][Gemma 4] Clamp soft-token estimate to max_soft_tokens

合并时间：2026-05-02 14:34

原文链接：http://prhub.com.cn/vllm-project/vllm/pull/40796

执行摘要

- 一句话：约束 Gemma 4 软 token 估计值不超过 max_soft_tokens
- 推荐动作：建议阅读，了解多模态 token 计数一致性的重要性和最小修复策略。clamp 虽简单但正确，避免过度工程。

功能与动机

PR 描述指出：对于极端宽高比图像（如 3x900），_compute_num_soft_tokens 返回的 soft tokens 数量超过了 HF 图像处理器视觉塔实际输出的 max_soft_tokens，导致 placeholder 与编码器输出不匹配，引发 ValueError 崩溃。修复通过 clamp 确保 prompt 侧估计值不超过该上限。

实现拆解

步骤：

1. 修改 _compute_num_soft_tokens 方法，将返回值由 num_patches // (pooling_kernel_size**2) 改为 min(计算值, max_soft_tokens)。该 clamp 仅影响极端比例场景。
2. 在测试文件中新增回归测试 test_compute_num_soft_tokens_does_not_exceed_max_soft_tokens，使用 @pytest.mark.parametrize 覆盖四组参数。
3. 根据 review 讨论，移除了初始提交中包含的自定义异常和引擎层改动，仅保留 clamp 修复。

关键文件：

- vllm/model_executor/models/gemma4_mm.py（模块 Gemma4；类别 source；类型 core-logic；符号 _compute_num_soft_tokens, Gemma4ProcessingInfo）：核心修复文件，修改 _compute_num_soft_tokens 方法添加 clamp 以防止 soft token 计数超出上限。
- tests/models/multimodal/processing/test_gemma4.py（模块 测试；类别 test；类型 test-coverage；符号 test_compute_num_soft_tokens_does_not_exceed_max_soft_tokens）：新增回归测试，覆盖 clamp 场景，确保估算值不超过上限。

关键符号：_compute_num_soft_tokens

关键源码片段

vllm/model_executor/models/gemma4_mm.py

核心修复文件，修改 _compute_num_soft_tokens 方法添加 clamp 以防止 soft token 计数超出上限。

```
# vllm/model_executor/models/gemma4_mm.py (Gemma4ProcessingInfo._compute_num_soft_
tokens)
def _compute_num_soft_tokens(
    self,
    image_width: int,
    image_height: int,
    max_soft_tokens: int | None = None,
) -> int:
    vision_cfg = self.get_hf_config().vision_config
    patch_size = vision_cfg.patch_size
    pooling_kernel_size = vision_cfg.pooling_kernel_size

    if max_soft_tokens is None:
        max_soft_tokens = vision_cfg.default_output_length

    unit = patch_size * pooling_kernel_size
    max_patches = max_soft_tokens * pooling_kernel_size**2
    num_patches_orig = (image_height / patch_size) * (image_width / patch_size)
    scale = math.sqrt(max_patches / num_patches_orig)
    target_h = max(unit, int(math.floor(image_height * scale / unit)) * unit)
    target_w = max(unit, int(math.floor(image_width * scale / unit)) * unit)
    num_patches = (target_h // patch_size) * (target_w // patch_size)
    # Clamp to ``max_soft_tokens``: 极端宽高比（例如 3x900）会导致
    # floor() 将某个维度向上舍入到 ``unit`` 而另一维度自由缩放，
    # 超出 ``max_patches``。HF Gemma 4 图像处理器将其视觉塔输出
    # 上限设为 ``max_soft_tokens``，若无此 clamp，prompt 端占位符
    # 数量超过编码器输出，引发崩溃。
    return min(num_patches // (pooling_kernel_size**2), max_soft_tokens)
```

评论区精华

主要讨论点：

- DarkLight1337 指出多模态处理器应在进入引擎前确保占位符正确，无需在引擎层捕获错误。
- Isotr0py 建议不要引入模型特定异常，直接复用 ValueError。
- 测试方式：Isotr0py 建议使用模型上下文构建 *e2e* 测试，而非 mock；作者随后调整。 最终决定采用最小修复方案，仅 clamp，无引擎修改，无自定义异常。
- 错误处理层级：*engine* 层 vs processor 层 (design): 采用最小修复方案，只修处理器，不涉及引擎。
- 是否引入自定义异常 (design): 放弃自定义异常，保持简单。
- 测试方式：mock 方式 vs 模型上下文方式 (testing): 采用 *e2e* 风格测试，通过 MULTIMODAL_REGISTRY 创建 processor。

风险与影响

- 风险：风险极低。clamp 是对已有上界的严格收紧，任何此前返回 ≤ max_soft_tokens 的调用不受影响；此前返回 > max_soft_tokens 的调用已崩溃，现在被预防。唯一潜在风险是如果 max_soft_tokens 配置有误，但那是外部配置问题。
- 影响：影响范围限于 Gemma 4 模型的多模态推理，特别是处理极端宽高比图像。用户将不再遇到 Attempted to assign … multimodal tokens to … placeholders 导致的引擎崩溃。向后兼容，无 API 变更。
- 风险标记：极端宽高比修复，向后兼容，新增测试覆盖

关联脉络

- PR #40599 [MM][Gemma4] Support max_soft_tokens="auto": 同一 Gemma4 多模态处理模块，解决不同但相关的 max_soft_tokens 使用问题。
- PR #40347 [Bugfix][Gemma4] Fix vision fp16 overflow causing output: 同一模型的另一项 bugfix，修复视觉 fp16 溢出。