

PR #49015 完整报告

vllm-project/vllm

[Bugfix] Qwen3-VL/Qwen-Omni: honor max_pixels/min_pixels for video prompts

合并时间: 2026-07-19 01:29

原文链接: <http://prhub.com.cn/vllm-project/vllm/pull/49015>

执行摘要

- 一句话: 修复 Qwen 视频 max_pixels 被丢弃的问题
- 推荐动作: 该 PR 值得精读, 尤其适用于理解 vLLM 多模态处理器的 mm_kwarg 传递与 HF 处理器之间的适配层机制。其设计决策——在 `_call_hf_processor` 内部通过 `ctx.get_merged_mm_kwarg` 统一读取引擎保存的配置, 而非修改外部过滤逻辑——值得作为类似问题的修复模式参考。

功能与动机

Fixes #48733. 用户报告 Qwen3-Omni 在线推理的 `prompt_tokens` 比 HF 处理器语义高出约 65%, 原因是 `mm_processor_kwarg` 中的 `max_pixels` 和 `min_pixels` 被静默丢弃, 导致视频使用未限制的原生分辨率。PR body 详细说明了两个根因: HF 视频处理器只从 `size.{longest,shortest}_edge` 读取帧像素预算; 引擎的 `get_allowed_kwarg_only_overrides` 过滤掉了不在 HF 签名中的 `max_pixels`。

实现拆解

1. Qwen3-VL 视频处理分支 (`vllm/model_executor/models/qwen3_vl.py`): 在 `_call_hf_processor` 的视频循环中, 新增 `merged = self.info.ctx.get_merged_mm_kwarg(mm_kwarg)`, 当 `merged` 包含 `size/min_pixels/max_pixels` 时, 读取 `self.info.get_video_processor().size` 作为基准, 将 `min_pixels` 映射到 `shortest_edge`, `max_pixels` 映射到 `longest_edge`, 合并后写入 `video_mm_kwarg["size"]`, 使 HF 处理器能够正确识别像素预算。
2. Qwen2.5-Omni/Qwen3-Omni 视频处理分支 (`vllm/model_executor/models/qwen2_5_omni_thinker.py`): 在 `_call_hf_processor` 中, 新增相同的合并逻辑, 但先检查 `mm_data.get("videos")` 非空, 然后通过 `self.info.get_hf_processor().video_processor.size` 获取处理器基准尺寸, 同样将 `min_pixels/max_pixels` 映射到 `shortest_edge/longest_edge`。
3. 无变更时的行为: 当 `mm_processor_kwarg` 中未传递 `size/min_pixels/max_pixels` 时, 两处逻辑均不触发, `mm_kwarg` 保持不变, 保持 `uncapped` 行为不变, 无回归风险。

关键文件:

- `vllm/model_executor/models/qwen2_5_omni_thinker.py` (模块 模型执行器; 类别 `source`; 类型 `data-contract`): 修复 Qwen2.5-Omni 和 Qwen3-Omni 视频 `max_pixels` 被丢弃的

核心变更文件。在 `_call_hf_processor` 中新增从 `ctx.get_merged_mm_kwargs` 读取并翻译 `size/min_pixels/max_pixels` 的逻辑。

- `vllm/model_executor/models/qwen3_vl.py` (模块 模型执行器; 类别 `source`; 类型 `data-contract`) : 修复 Qwen3-VL 视频 `max_pixels` 被丢弃的核心变更文件。在 `_call_hf_processor` 的视频循环中新增使用 `ctx.get_merged_mm_kwargs` 的像素预算处理逻辑。

关键符号: `Qwen2_5OmniThinkerMultiModalProcessor._call_hf_processor`,
`Qwen3VLMultiModalProcessor._call_hf_processor`

关键源码片段

`vllm/model_executor/models/qwen2_5_omni_thinker.py`

修复 Qwen2.5-Omni 和 Qwen3-Omni 视频 `max_pixels` 被丢弃的核心变更文件。在 `_call_hf_processor` 中新增从 `ctx.get_merged_mm_kwargs` 读取并翻译 `size/min_pixels/max_pixels` 的逻辑。

```
# vllm/model_executor/models/qwen2_5_omni_thinker.py
# 在 _call_hf_processor 的音频处理后、调用 super() 之前插入

merged = self.info.ctx.get_merged_mm_kwargs(mm_kwargs)
# 仅当有视频数据且用户指定了 size/min_pixels/max_pixels 才处理
if mm_data.get("videos") and (
    merged.keys() & {"size", "min_pixels", "max_pixels"}
):
    mm_kwargs = dict(mm_kwargs)
    # 从 HF 处理器获取默认的 video_processor.size 字典
    video_size = dict(self.info.get_hf_processor().video_processor.size)
    size_override = merged.get("size")
    if size_override is not None:
        video_size = video_size | size_override # 合并用户传递的 size 覆盖
    min_pixels = merged.get("min_pixels")
    if min_pixels is not None:
        video_size["shortest_edge"] = min_pixels # 映射 min_pixels -> shortest_edge
    max_pixels = merged.get("max_pixels")
    if max_pixels is not None:
        video_size["longest_edge"] = max_pixels # 映射 max_pixels -> longest_edge
    mm_kwargs["size"] = video_size # 替换 size 使 HF 处理器能识别像素预算
```

`vllm/model_executor/models/qwen3_vl.py`

修复 Qwen3-VL 视频 `max_pixels` 被丢弃的核心变更文件。在 `_call_hf_processor` 的视频循环中新增使用 `ctx.get_merged_mm_kwargs` 的像素预算处理逻辑。

```
# vllm/model_executor/models/qwen3_vl.py
# 在 _call_hf_processor 的视频循环中创建 video_mm_kwargs 后、设置 fps 前插入

video_mm_kwargs = dict(**mm_kwargs)
merged = self.info.ctx.get_merged_mm_kwargs(mm_kwargs)
```

```
# 当用户指定了 size/min_pixels/max_pixels 中的任一键时处理
if merged.keys() & {"size", "min_pixels", "max_pixels"}:
    # 从 HF 视频处理器获取默认 size (包含 shortest_edge 和 longest_edge)
    video_size = dict(self.info.get_video_processor().size)
    size_override = merged.get("size")
    if size_override is not None:
        video_size = video_size | size_override # 合并用户传递的 size 覆盖
    min_pixels = merged.get("min_pixels")
    if min_pixels is not None:
        video_size["shortest_edge"] = min_pixels # min_pixels -> shortest_edge
    max_pixels = merged.get("max_pixels")
    if max_pixels is not None:
        video_size["longest_edge"] = max_pixels # max_pixels -> longest_edge
    video_mm_kwargs["size"] = video_size # 替换 size 使 HF 处理器正确限制帧分辨率
```

评论区精华

无 review 评论。仅有一条来自 github-actions 的自动欢迎消息和一条来自 claude[bot] 的自动回复（提及 fork PR 无法自动审查）。最终由 ywang96 直接批准合并。

- 暂无高价值评论线程

风险与影响

- 风险：风险较低。变更仅在视频分支且显式检查 `merged.keys() & {"size", "min_pixels", "max_pixels"}`，非破坏性。但需注意：`min_pixels` 和 `max_pixels` 映射到 `shortest_edge/longest_edge` 是语义等价假设，若 HF 处理器未来改变这些键的含义可能失效。另外，`video_size` 从默认处理器尺寸构建，若用户同时传递了 `size` 和 `max_pixels`，`size` 先被合并，然后 `max_pixels` 会覆盖 `longest_edge`，行为符合预期。
- 影响：影响范围集中在 Qwen3-VL、Qwen2.5-Omni 和 Qwen3-Omni 三个模型系列。用户侧效果显著：PR body 中的 E2E 测试显示，对于 1280x720 视频，`max_pixels=393216` 时 `prompt_tokens` 从 12044 降至 324；对于 480x832 视频，`max_pixels=50176` 时从 4086 降至 486。这直接降低 TTFT 和显存占用，提升系统吞吐量。对不传递 `max_pixels` 的用户无影响。
- 风险标记：需要模型行为验证

关联脉络

- PR #43403 [Bugfix][Multi-Modality] Qwen3VL: Video ViT CUDA-graph fallback: PR body 提及该 PR 是与视频相关的另一个修复，但与此 PR 无关（仅用以说明不是重复 PR）。